(2)

A TRIDENT SCHOLAR PROJECT REPORT

NO. 178

A FRACTAL COMPUTER MODEL OF MACROMOLECULE-CELL SURFACE INTERACTIONS

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

A FRACTAL COMPUTER MODEL OF MACROMOLECULE-CELL SURFACE INTERACTIONS

A Trident Scholar Project Report

by

Midshipman Jeffrey S. Dodge, Class of 1991

U. S. Naval Academy

Annapolis, Maryland

Adviser: Associate Professor Boyd A. Waite
Chemistry Department

Accepted for Trident Scholar Committee

Francis D Correll
Chair

13 May 1991
Date

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	13 May 1991	Final 1990/91

4. TITLE AND SUBTITLE

A FRACTAL COMPUTER MODEL OF MACROMOLECULE-CELL SURFACE INTERACTIONS

5. FUNDING NUMBERS

6. AUTHOR(S)

Jeffrey S. Dodge

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Naval Academy, Annapolis, MD

8. PERFORMING ORGANIZATION REPORT NUMBER

U.S.N.A. - TSPR; 178 (1991)

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Accepted by the U.S. Trident Scholar Committee.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

This document has been approved for public release; its distribution is UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Macromolecules involved in binding to cell surface receptors are important in many biological systems. The cross-linking and clustering of receptors that an antigen can cause is a vital event in the activation of the B-lymphocyte cell and the subsequent initiation of the immune response. This project set out to create a computer model of the interactions between a large molecule with several binding sites, and receptors on the cell surface. In order to have some correlation with a real system, a multivalent antigen and a B-cell were used for physical parameters. The model was designed to provide insight into the behavior of the system, and information on the configurations of bound macromolecules. Analysis of the results from the computer model took two forms, interpretation of the visual output, and numerical analysis of the bound receptor clusters. Fractal geometry was used to characterize the shapes of the cross-linked cell receptors.

14. SUBJECT TERMS

Macromolecules, Cell membranes, Immunology, Cell receptors, Fractals

15. NUMBER OF PAGES

80

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	

ABSTRACT

Macromolecules involved in binding to cell surface receptors are important in many biological systems. The cross-linking and clustering of receptors that an antigen can cause is a vital event in the activation of the B-lymphocyte cell and the subsequent initiation of the immune response. Experiments show that cross-linking is needed to activate immune cells, but the mechanism of the process, and the characteristics of the cluster formed are unknown.

This project set out to create a computer model of the interactions between a large molecule with several binding sites, and receptors on the cell surface. In order to have some correlation with a real system, a multivalent antigen and a B-cell were used for physical parameters. The model was designed to provide insight into the behavior of the system, and information on the configurations of bound macromolecules.

The model made extensive use of the graphics capability of the computer used, a Silicon Graphics Personal Iris workstation. This gave the program greater flexibility by allowing the investigator to visually inspect the system in action, gaining insight into an otherwise unobserved process.

Analysis of the results from the computer model took

two forms, interpretation of the visual output, and numerical analysis of the bound receptor clusters. The visual output addressed several concerns that are present in current theory on the system. Findings on the free medium motion of the macromolecule, as well as the entropy, rate, and spatial configurations of the molecule-receptor cluster were all compared to current assumptions. Fractal geometry was used to characterize the shapes of the cross-linked cell receptors.

The results represent only a low order utilization of the model, and with further refinement this computer model might reveal significant facts about the action of cell receptors. The program architecture is flexible, and encourages such development.

TABLE OF CONTENTS

I. The Immune System

The immune system provides organized and specific defense responses that protect against infection with pathogenic microorganisms and against the development and spread of malignant tumors. The system recognizes foreign bodies and reacts with cytotoxic cells, macrophages and related cell types that ingest foreign agents, and the lysing proteins of the complement system. The foreign molecules, cells and viruses which activate the immune response are known as antigens since they act to generate an immune response. (1) The cells in the immune system that are responsible for the activation of the immune response are known as lymphocytes. These lymphocytes are individually specialized in their commitment to respond to a limited group of structurally related antigens. This commitment exists prior to the system's first exposure to a given antigen, and is due to the presence on the lymphocyte membrane of binding sites or receptors which are specific to determinants on that type of antigen. (2)

Besides the differences in the specificity of their binding sites, lymphocytes also differ in their functional properties. Lymphocytes can be divided into two broad functional classes: B-cells, which mature in the bone marrow, and T-cells which mature in the thymus.

T-lymphocytes are responsible for the cell-mediated immune response (Figure 1), and consist of a large series

Cell-Mediated Immune Response

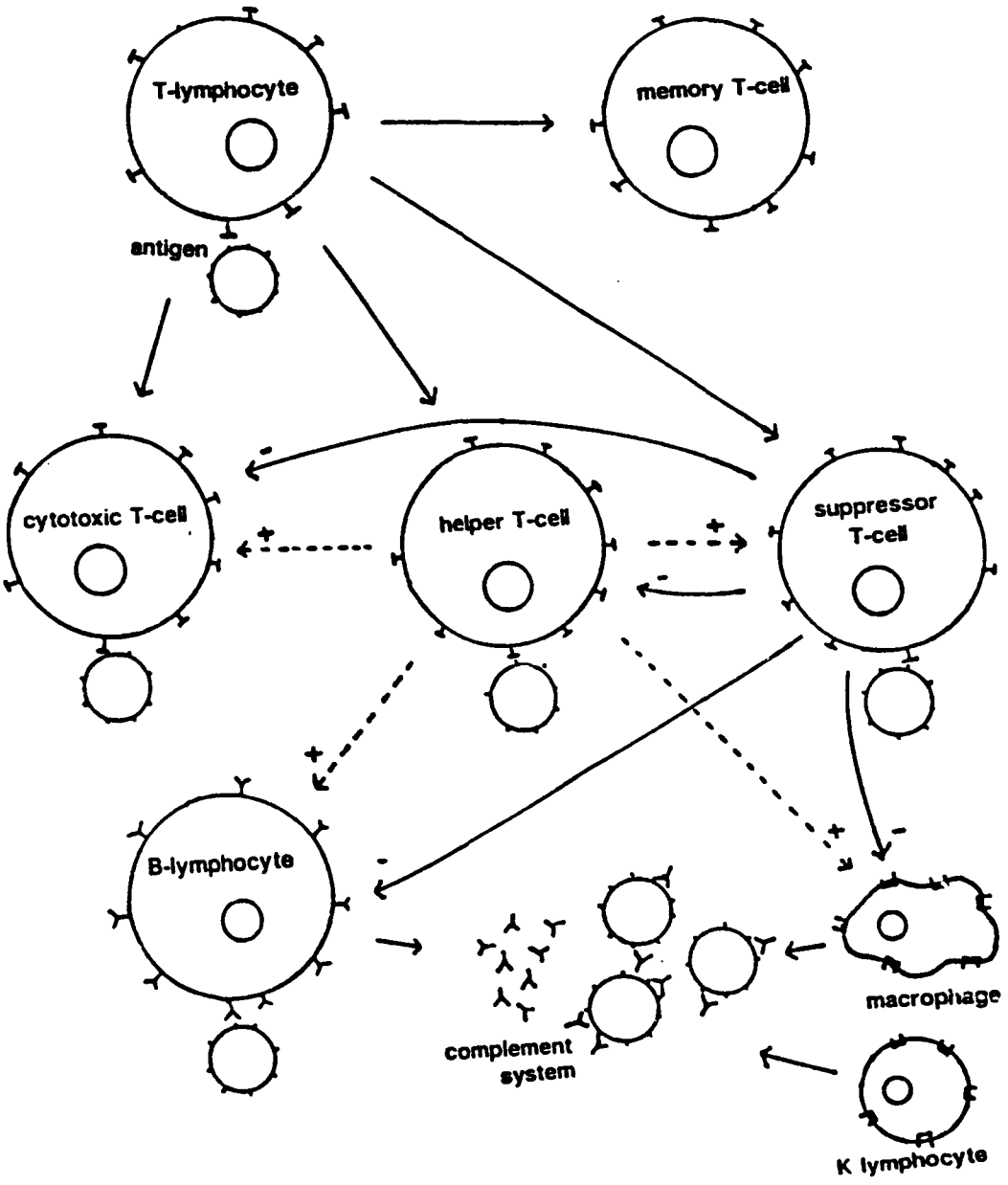


Figure 1

of subtypes. Some T-lymphocytes mediate important regulatory functions, such as the ability to help or suppress the development of immune responses, while others are involved in effector functions, such as the production of soluble products that initiate a variety of inflammatory responses, or the direct destruction of agents bearing antigenic substances. These responses are generally effective against larger antigens, such as parasites, cancer and infected cells. T-cell response is important, but the complexity of its interactions makes it difficult to model. The B-cell response and activation is easier to control and analyze.

B-lymphocytes move freely through the blood and lymph and are the source of the humoral immune response. (Figure 2) When a B-cell encounters an antigen that stimulates its receptors, it will become activated when it also encounters the proper soluble growth factors or signals from helper T-cells. The B-lymphocytes undergo proliferation and differentiation in response to the activating antigen. Proliferation increases the number of cells available for activation, and provides a larger number of B cells so that a second encounter with the same antigen evokes a response of greater magnitude and promptness than the primary response. As the graph in Figure 3 shows, the increased number of B cells means an increase in the strength and speed of the immune response. These precursor (unactivated) B cells are termed memory cells since they

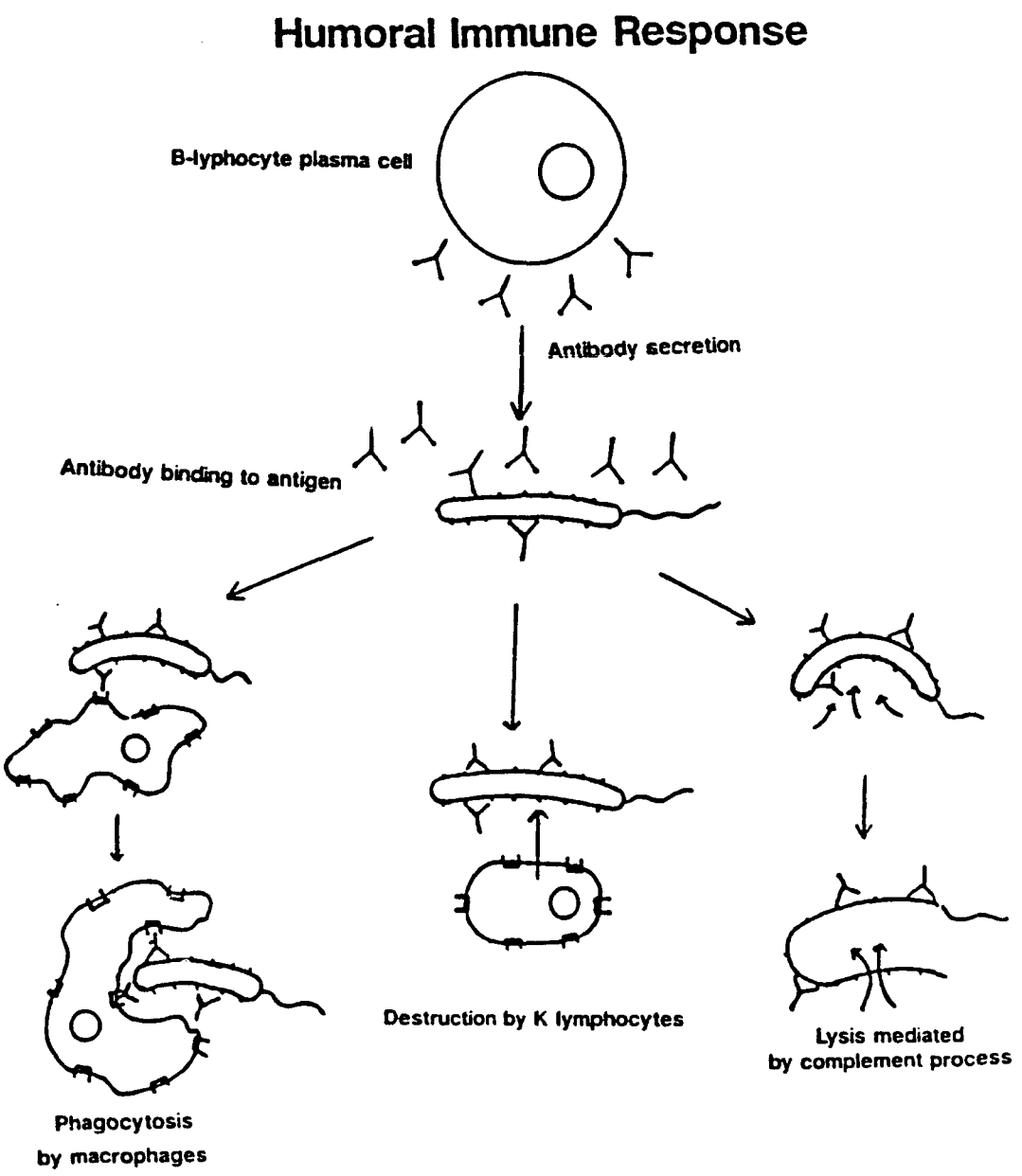

Humoral Immune Response
B-lyphocyte plasma cell
Antibody secretion
Antibody binding to antigen
Phagocytosis
by macrophages
Destruction by K lymphocytes
Lysis mediated
by complement process

Figure 2

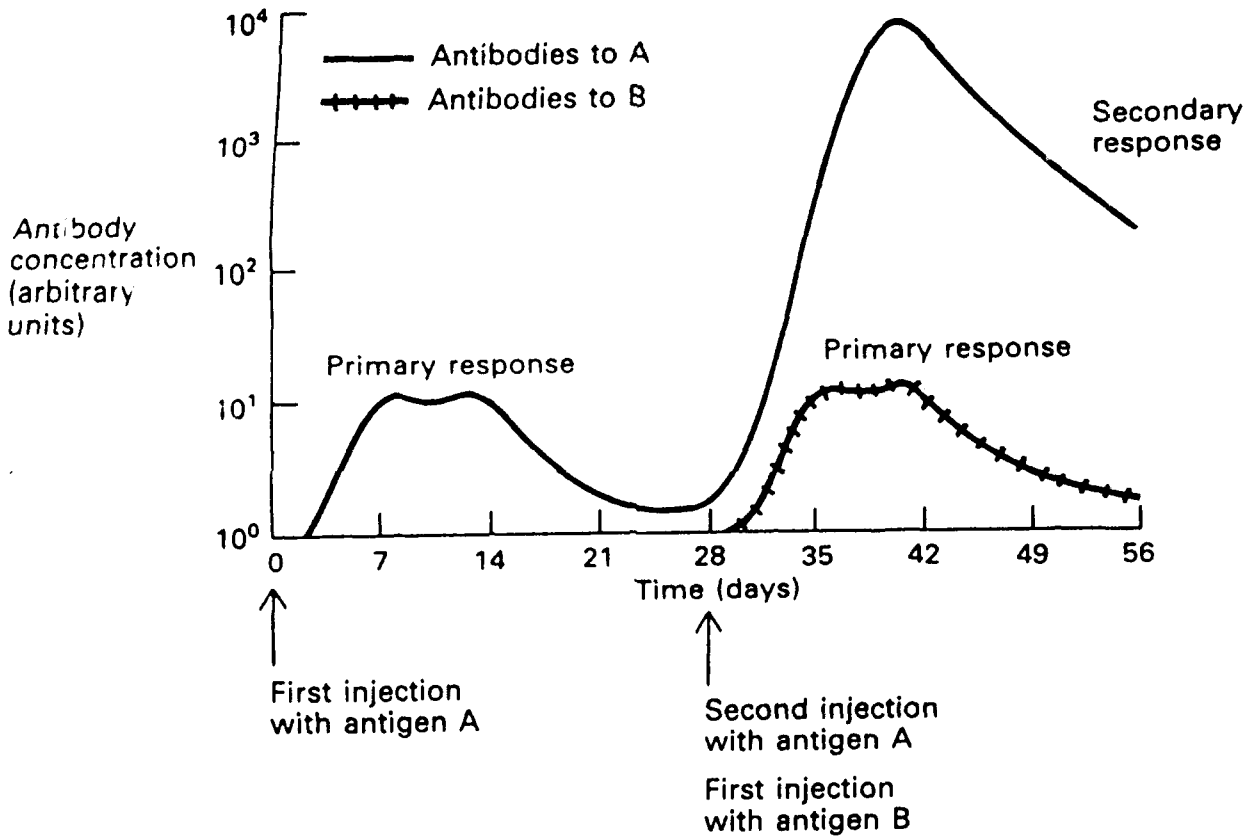


Figure 3. The increased response to the second injection of antigen is due to immunological memory.

lead to a state of immunological memory. (3) Some of the proliferated clones differentiate into antibody-secreting cells, the most common being plasma cells. These antibodies are the basis of the humoral response and will bind with the antigen and mark it for further immunologic action. The marked antigens are thus targeted for phagocytosis by macrophages, attack by cytotoxic lymphocytes, or lysis by the complement system.

The cell receptors on the B cell, and the antibodies B cells produce, are immunoglobulin molecules. (Figures 4,5) Immunoglobulins, or Igs, are a group of structurally similar proteins. They consist of a number of units, each of which is made up of two heavy (H) polypeptide chains, and two light (L) polypeptide chains. Each unit has two combining sites, or Fab fragments, that can bind to antigens. Both the light and heavy chains have a variable (V) region at the combining site that is different for individual immunoglobulins, giving specificity to the immune response. The remainder of the molecule is referred to as the constant (C) region and is responsible for the biological function of a class of antibodies.

There are five classes of antibodies; IgM, IgD, IgG, IgA, and IgE. IgM antibodies activate the complement system, IgAs are secreted in a variety of bodily fluids, IgEs crosslink receptors on mast cells and basophils with antigens, and IgDs act as membrane receptors. IgD, IgG, and IgE antibodies usually consist of a single unit of two

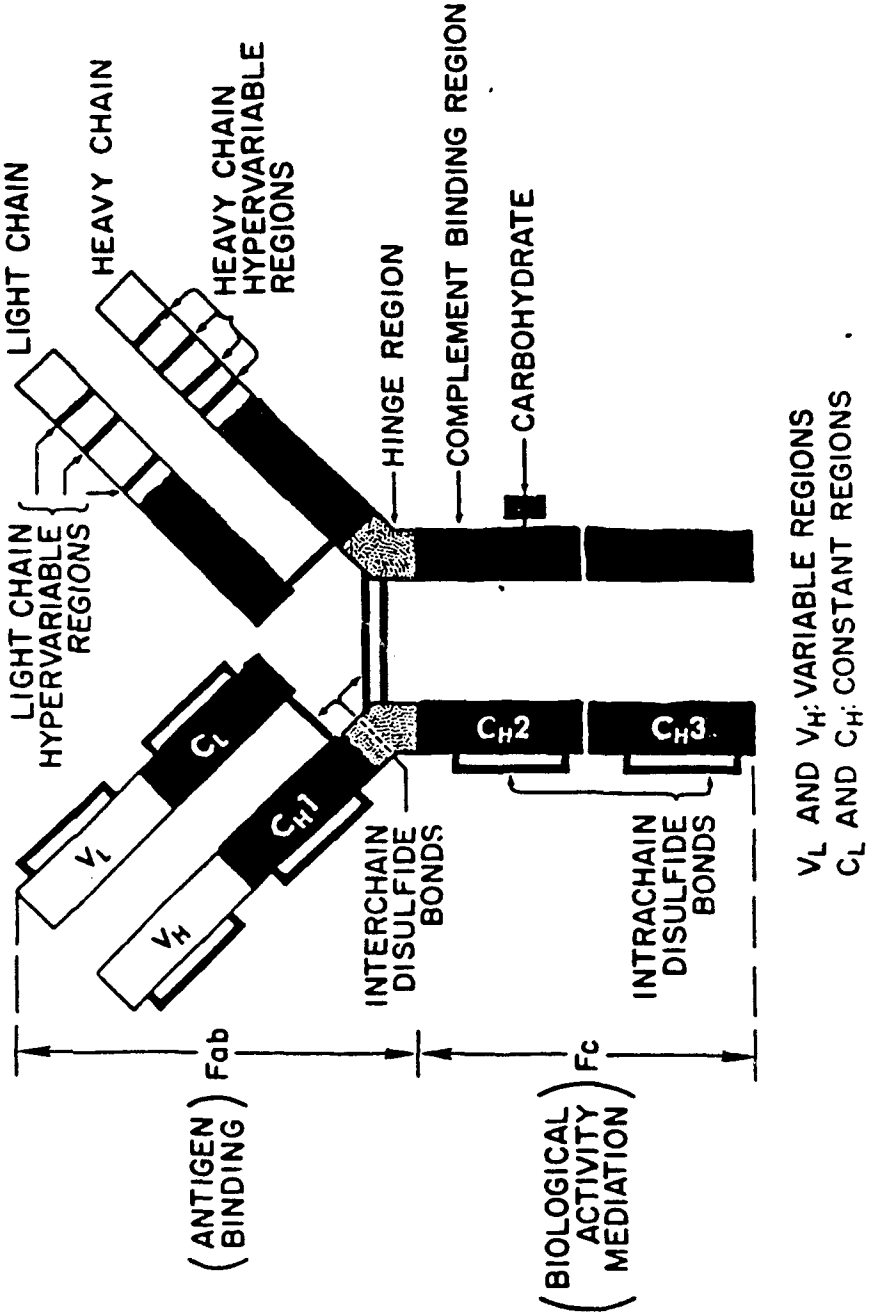

LIGHT CHAIN
HEAVY CHAIN
HEAVY CHAIN HYPERVARIABLE REGIONS
HINGE REGION
COMPLEMENT BINDING REGION
CARBOHYDRATE
LIGHT CHAIN HYPERVARIABLE REGIONS
C_L
V_L
C_H1
V_H
C_H2
C_H3
INTERCHAIN DISULFIDE BONDS
INTRACHAIN DISULFIDE BONDS
Fab
(ANTIGEN BINDING)
Fc
(BIOLOGICAL ACTIVITY MEDIATION)
V_L AND V_H: VARIABLE REGIONS
C_L AND C_H: CONSTANT REGIONS

Figure 4

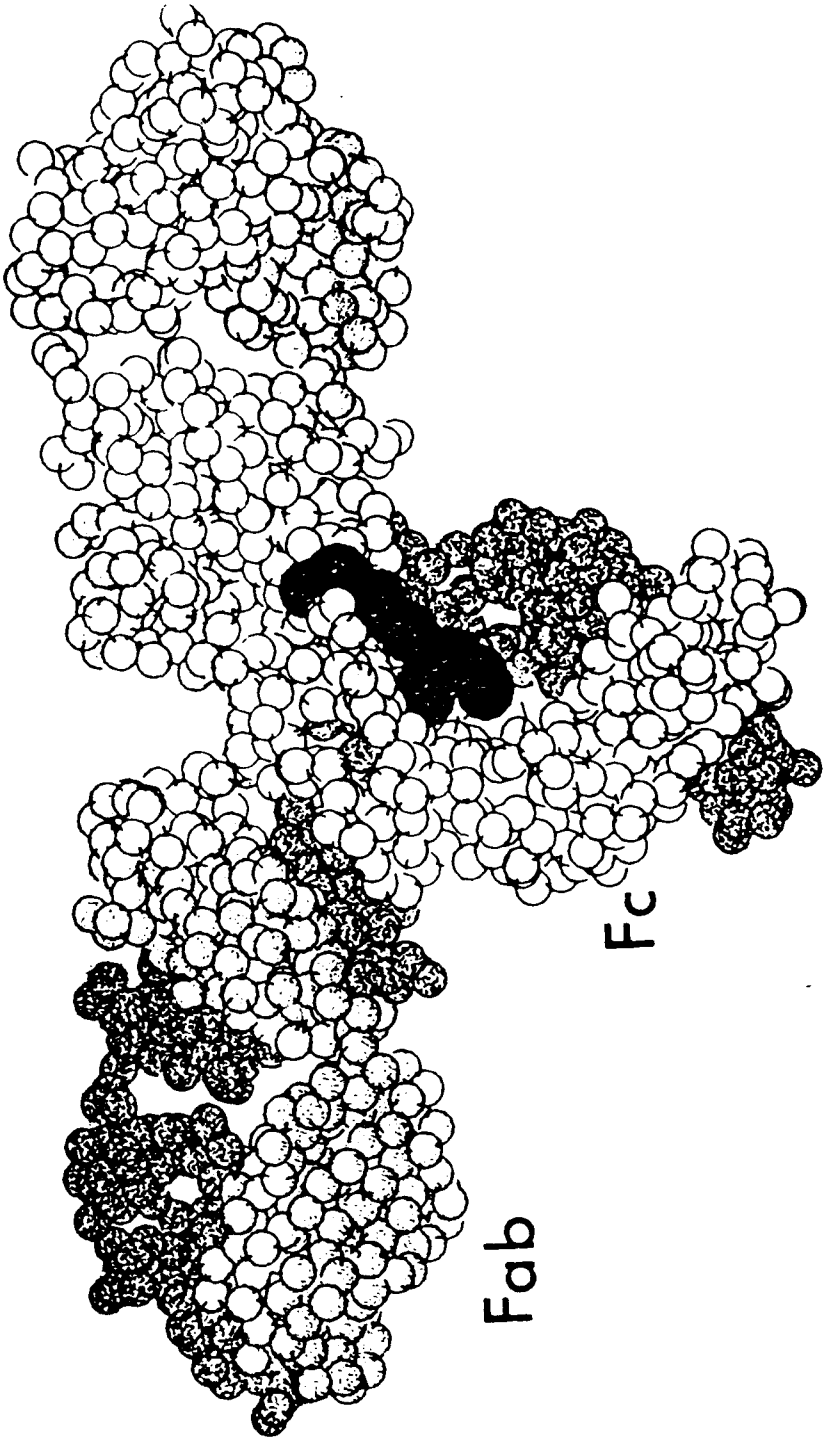


Figure 5. Space-filling model of Immunoglobulin

H and two L chains, while IgMs are made of five such units, although they are a single unit when they act as membrane receptors. Since each unit has two variable region binding sites, most antibodies are bivalent. IgM antibodies are decavalent since they are composed of five bivalent units. Valency is determined for antibodies and antigens by counting the number of binding sites on the molecule. Multivalent molecules often have functional valencies less than their total valency, since some binding sites may be inaccessible due to steric hindrance. (4)

The cell receptors differ from antibodies in the structure of the constant region. Free antibodies have a hydrophilic constant region, while the constant region tails of receptors are hydrophobic. This feature keeps the base of the cell receptor imbedded in the plasma membrane of the cell. The presence of hydrophilic binding sites makes it almost impossible for the receptor to flip over so that these sites are inside the cell. The combination of hydrophobic and hydrophilic regions acts to keep the axis of the cell receptor perpendicular to the surface of the cell.

The fact that it is the hydrophobicity of the cell receptor that keeps it attached to the cell is the cause of some of the receptor's interesting characteristics. The immunoglobulin is not constrained by attachment to a fixed cell structure, and is only limited by its preference for remaining inside the bi-layer cell membrane. This allows

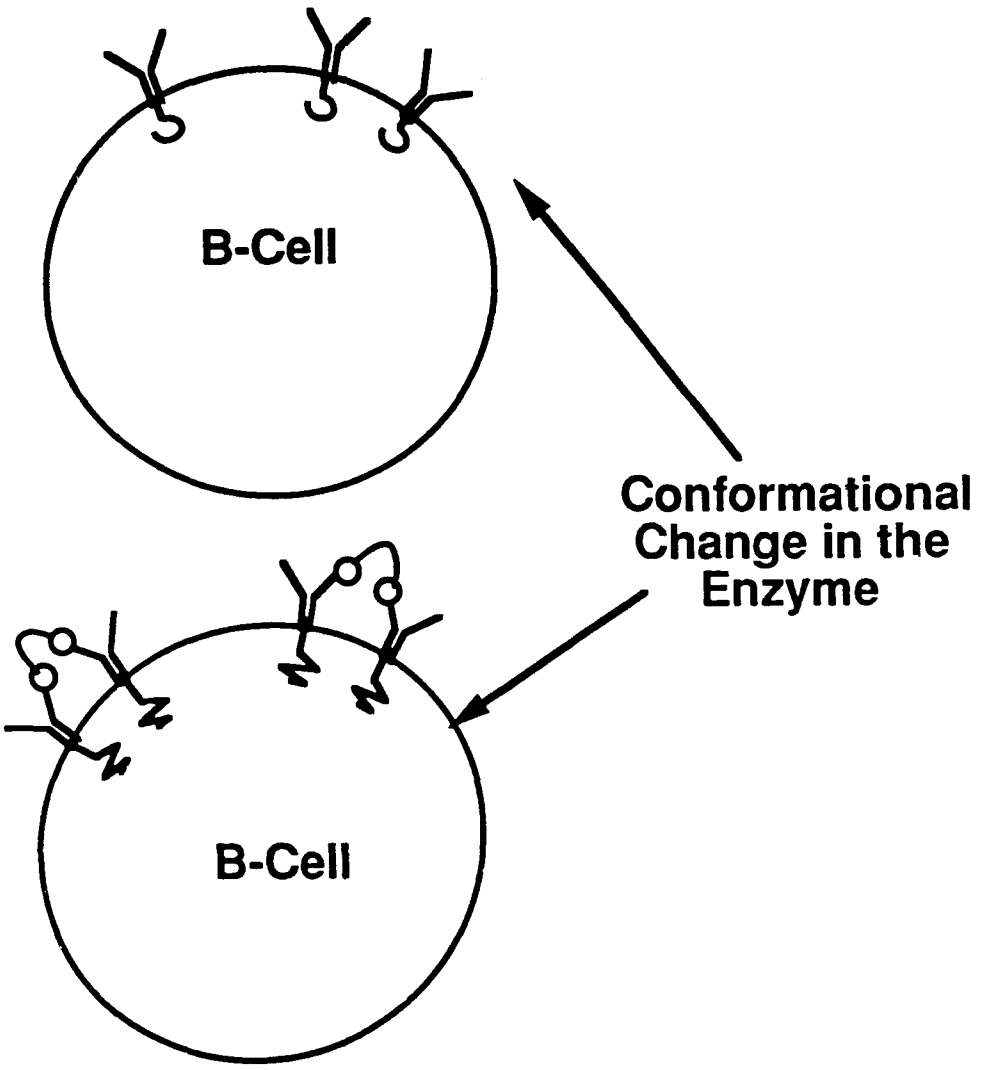
the cell receptor to move across the surface of the cell, conducting a two-dimensional random walk on the surface of a spheroid. The lack of a permanent binding site for the receptor allows a portion of the molecule to penetrate through the cell membrane into the interior of the cell. This segment of the molecule undergoes a conformational change when the receptor binds to an antigen (Figure 6), thus inducing conformational changes in the intramembrane enzymes associated with the receptor. These changes initiate the chemical pathway that leads to the cellular immune response. Figure 7 illustrates the chemical activity that occurs in the B cell after surface receptors have been cross-linked by an antigen.

2. Cross-Linking

In order to induce the enzymatic changes in the cell membrane, and achieve lymphocyte activation, receptors must be cross-linked into a receptor cluster. Cross-linking can occur when a multivalent antigen binds to more than one surface immunoglobulin. Figure 8 illustrates the sequential binding of an antigen to several cell surface receptors.

There is abundant experimental evidence that supports this idea. Mitogens, which are substances that activate cells and induce them to divide, are divalent or multivalent, and lose their mitogenicity when made

SURFACE IMMUNOGLOBULIN (s-Ig)



Antigens Cross-link s-Ig Receptors

Figure 6

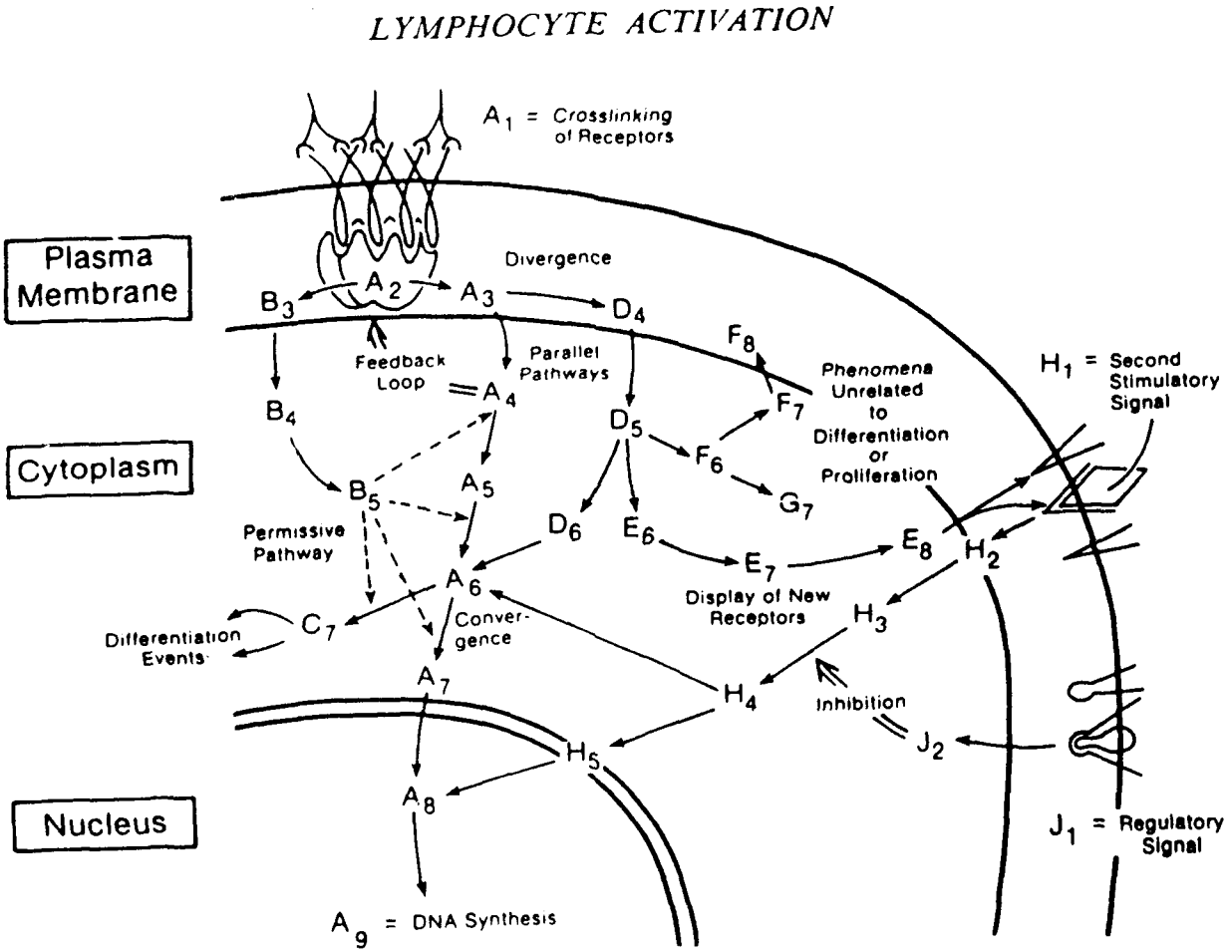


FIG. 7 A theoretical construct of the activation pathway showing how the various consequences of signals generated at the cell membrane might relate to one another. The sequence of activation events is shown as *single arrows;* inhibitory activities are shown as *double arrows.*

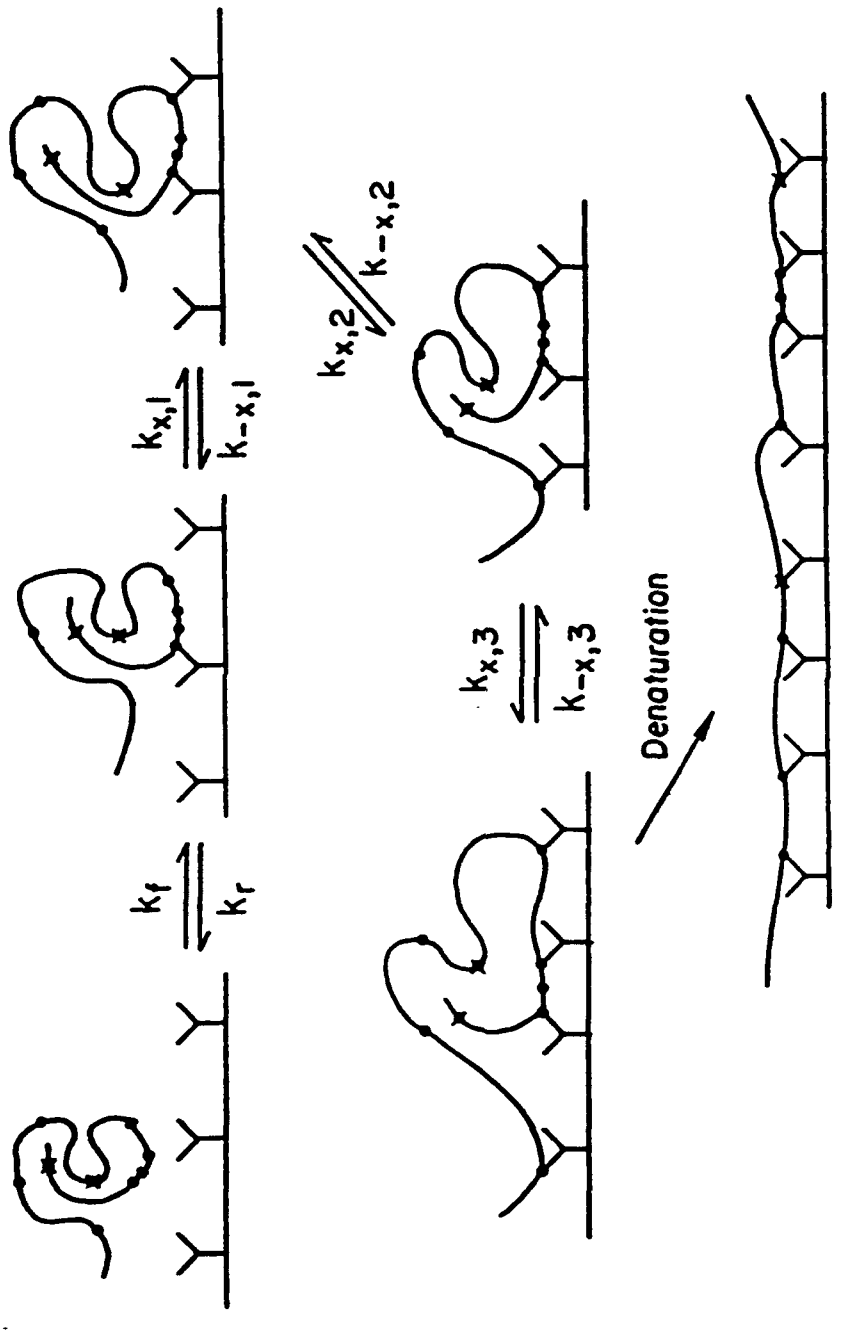


Figure 8. The binding of an antigen to a cell, showing various steps and rate constants.

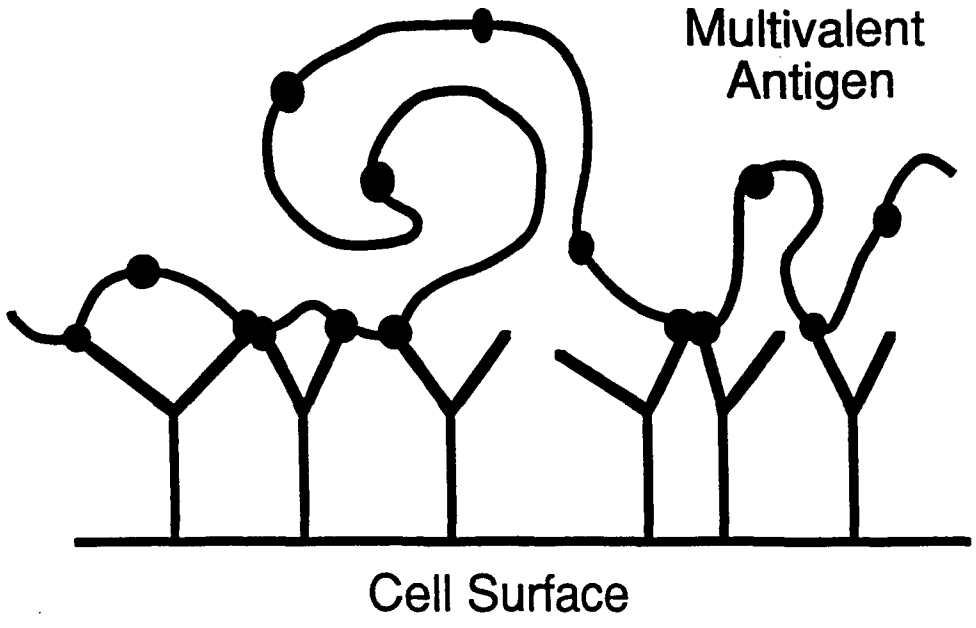
univalent. It is theorized that the few antigens that are univalent are functionally multivalent due to their interaction with T-helper factors and antigen presenting cells. Two experiments provide clearer evidence for cross-linking. The first is the use of anti-immunoglobulin antigen to stimulate B-cells. (5) Anti-immunoglobulin antigen, or anti-Ig is a molecule that binds to the constant Fc regions of surface immunoglobulin receptors. Bivalent anti-Ig can stimulate B-cells to undergo the first steps of activation, while monovalent Fab fragments of the same molecule do not elicit a response. Monovalent determinants that can engage a T-cell are also nonimmunogenic. Since at least two binding sites are required to activate the cell, cross-linking between receptor sites is indicated. The second finding that supports the cross-linking theory is the "high dose-low dose" phenomenon. (6) Extremely low doses and extremely high doses of an immunogenic substance do not elicit as great an immune response as doses between the two limits. This correlates with the cross-linking theory since at very low concentrations, there would not be enough antigenic material to cross-link surface immunoglobulins, while at very high concentrations, each binding site would be occupied by a different antigen, thus preventing a single antigen from cross-linking receptors. This explains the observed "high dose-low dose" trend in immune responses.

While it is clear that cross-linking of the surface

immunoglobulin cell receptors is a key step in the initiation of the immune response, the manner in which the receptors are cross-linked, and the configuration of an activating cluster are unknown. Current immunological thought holds that each lymphocyte cell is specific to a type of antigen, and will only be activated by that kind of antigen. This is known as the clonal selection theory since the activated cells are "selected", and proliferate to provide clones for increased response and immunological memory. (7) Since each cell is specific to only one type of antigen, all the receptors on that cell are specific to that same antigen. Thus all the receptors can be involved in the cross-linking. Experiments suggest that there is some critical number of receptors that must be linked into a cluster before the cell is activated, but this number, and the shape of the cross-linking, are unknown.

There are currently two competing theories regarding the formation of receptor clusters: the immunon theory and the low valence cluster theory. (Figure 9) Dintzis et al. propose that cross-linking is achieved by the binding of a single multivalent antigen to several surface immunoglobulins on the cell. (8) The antigen is usually thought of as a backbone of some kind, like the surface of a virus or an organic polymer, and has binding sites, or epitopes, spaced along its length. These epitopes are structural units that are able to bind with surface immunoglobulins, and are called haptens when in the free

Immunon Theory



Low Valence Theory

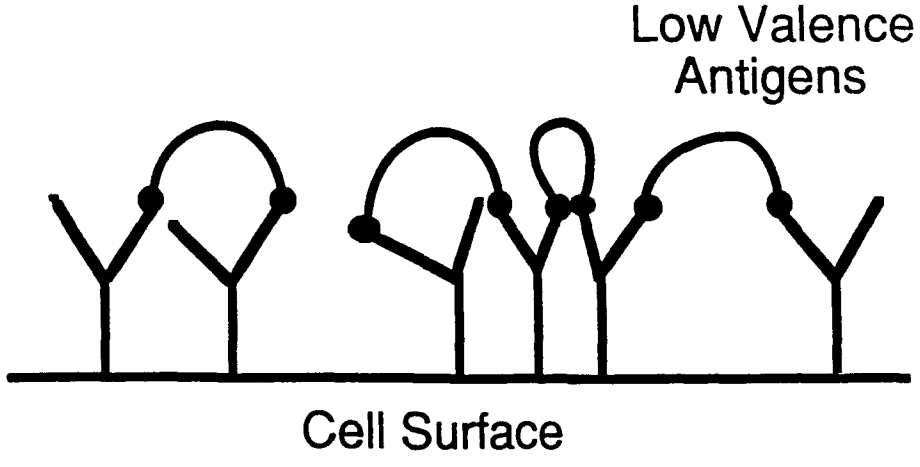


Figure 9. Two theories of cluster formation.

state, while when the haptens bind the antigen to receptors, the bound antigen-receptor cluster is called an immunon. These terms are illustrated in Figure 10. Experiments with a linear polyacrylamide chain haptenated with dinitrophenyl groups indicate that a threshold number between 10 and 20 effective epitopes is required for the molecule to be stimulatory to B-lymphocytes. Molecules with fewer epitopes were not immunogenic, and proved to be tolerogenic. This is to be expected since if the molecule can't form the critical size cluster it will not activate the cell, but it can still occupy receptors and prevent them from binding with a stimulatory antigen. The determination of effective epitope number is not clear, but the evidence does point to a threshold number of epitopes being required for B-cell stimulation.

The second theory, as espoused by Perelson and others, holds that bi- or trivalent antigens are stimulatory. (9) Experiments show that bivalent anti-immunoglobulin antigens cross-link the constant regions of surface receptors, and can start the activation of the lymphocyte. (10) Low valence clustering has also been seen in other immune cells, namely basophils and mast cells. (11) The theory holds that an antigen and receptor bind at a single site, leaving an open site on both the receptor and the antigen. Other receptors and antigens bind to these open sites, forming ring-like or chain-like clusters. These clusters perform the function of an immunon, but are

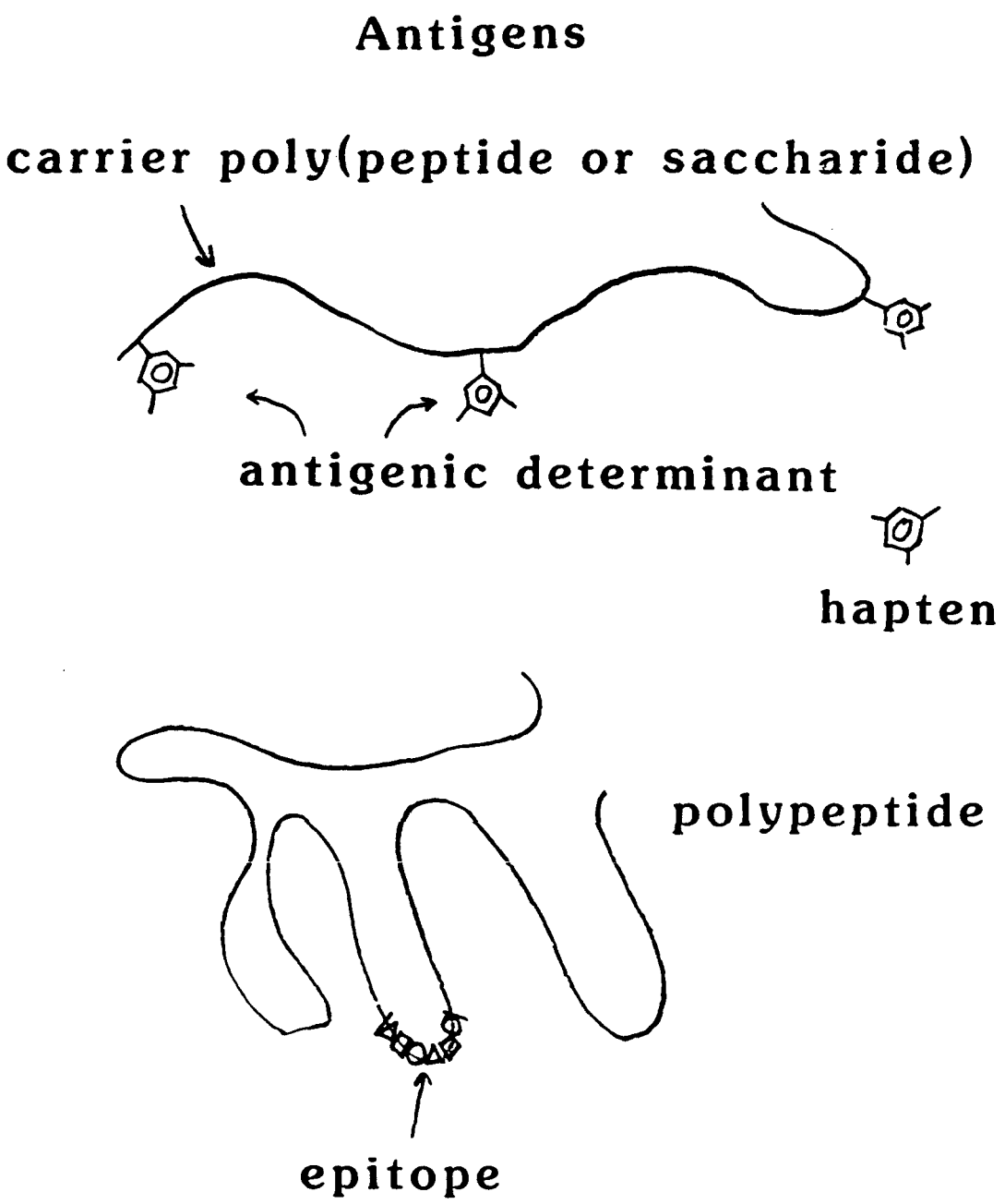


Figure 10. Terms used in describing antigens.

made up of several antigens and receptors. This idea that any antigen with a valency greater than two can evoke a response is theoretically sound and logical given the current understanding of the system.

This apparent contradiction between differing experimental results, and competing theories has led Perelson to call for a theory that "explicitly incorporates monogamous bivalent attachment of receptors to antigens and concerns itself with making detailed predictions of properties of receptor clusters (e.g., their size, shape, and mobility) as a function of the concentration and properties of the polymeric antigen (e.g., the number of haptens, their spacing, the flexibility of the backbone, and electrostatic, hydrophobic, and other interactions between the backbone and the surface." (12) An accurate theory would be of great benefit, for by increasing the understanding of the immune system, it would allow greater improvement in the treatment of immune diseases, and the enhancement of immune response. But before the theory can be formulated, the experimental evidence must be reevaluated, and the conceptual basis for competing arguments must be understood. A theoretical and conceptual basis of current evidence is needed in order to create a comprehensive theory. This task is initiated by this project, which models the system as it is understood today in the hopes that the model will increase the accuracy and effectiveness of the current theory.

3. Fractal Applications

Fractal geometry is a field of mathematics that can be used to provide descriptions of irregular forms in nature. (13) From coastlines to crystals, any complex object, especially random ones, can be described in terms of a non-integer dimension. This non-integer, or fractional, dimension is also called a fractal dimension, and is the basis for fractal mathematics. The existence of a fractal dimension can be explained by the examination of a random walk. A random walk on a two dimensional plane would be a straight line at any instant, thus having an instantaneous dimension of one. If the moving object continued its random walk to infinity, its track would eventually pass over every spot in the plane, thus giving the walk a dimension of two after an infinite time. However, at any finite time, the random walk is not a straight line, but it does not completely fill the plane. It is therefore said to have a dimension between one and two, which necessarily gives it a fractal dimension. The same analysis holds true for a random walk in three dimensions, giving a fractal dimension between one and three after a finite period of time. Figure 11 shows space-filling representations of a line , a surface and a two-dimensional fractal object. The graphs show that when measured with a unit length, the fractal object does not really extend out in two dimensions, but it covers more

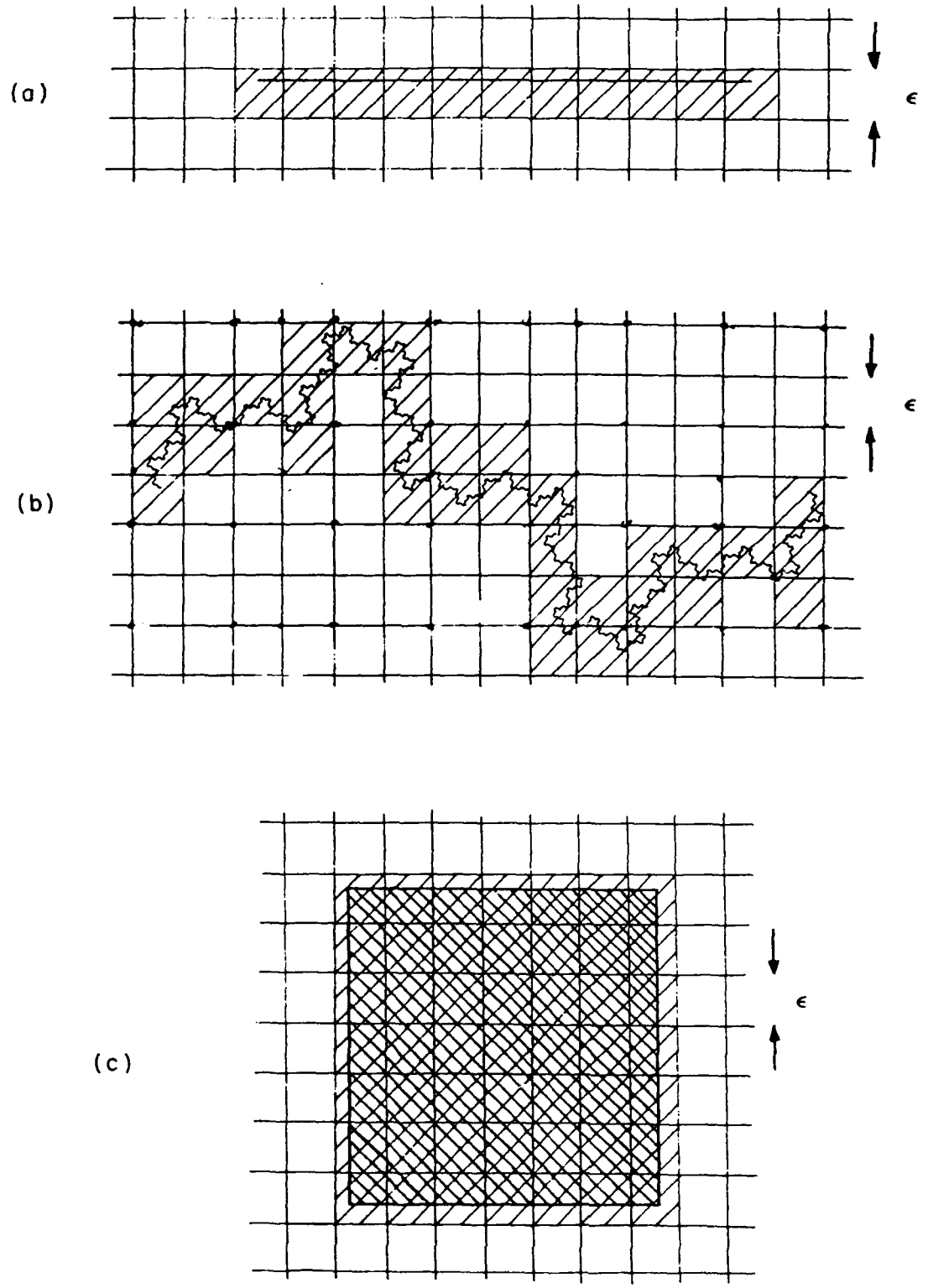


Figure 11. Graphs of a line (a), fractal line (b), and square (c), showing the dimensionality of each.

area than a straight line. This is another example of a fractal dimension.

One of the important properties of fractals, besides their irregularity, is their high level of disorder. (14) This means that the object is disordered at all length scales. Expansion of a piece of a fractal object yields a picture that looks just as disordered as the original picture. In fact the disorder will be at the same level for the whole object and for a piece of that object. Natural fractal objects differ from their ideal counterparts in that their level of disorder is not the same at all length scales. A natural object will have both an upper and lower size limit beyond which the object is not fractal. At any length scale between these size limits, or cutoffs, the natural object displays the properties of a fractal, while outside of the cutoffs the fractal description falls apart. Diffusion near a cell surface is a fractal process as shown in Figure 12. The various length scales used, from the atomic to the cellular, all show the same random motion. This disordered hierarchy is indicative of fractals.

The fact that fractals describe systems with a high level of disorder makes them excellent tools for dealing with irregular shapes. The use of fractal dimension allows comparison between irregular and vastly different objects. Similar dimensionality for the different objects can provide insight into the similarity of the forces that

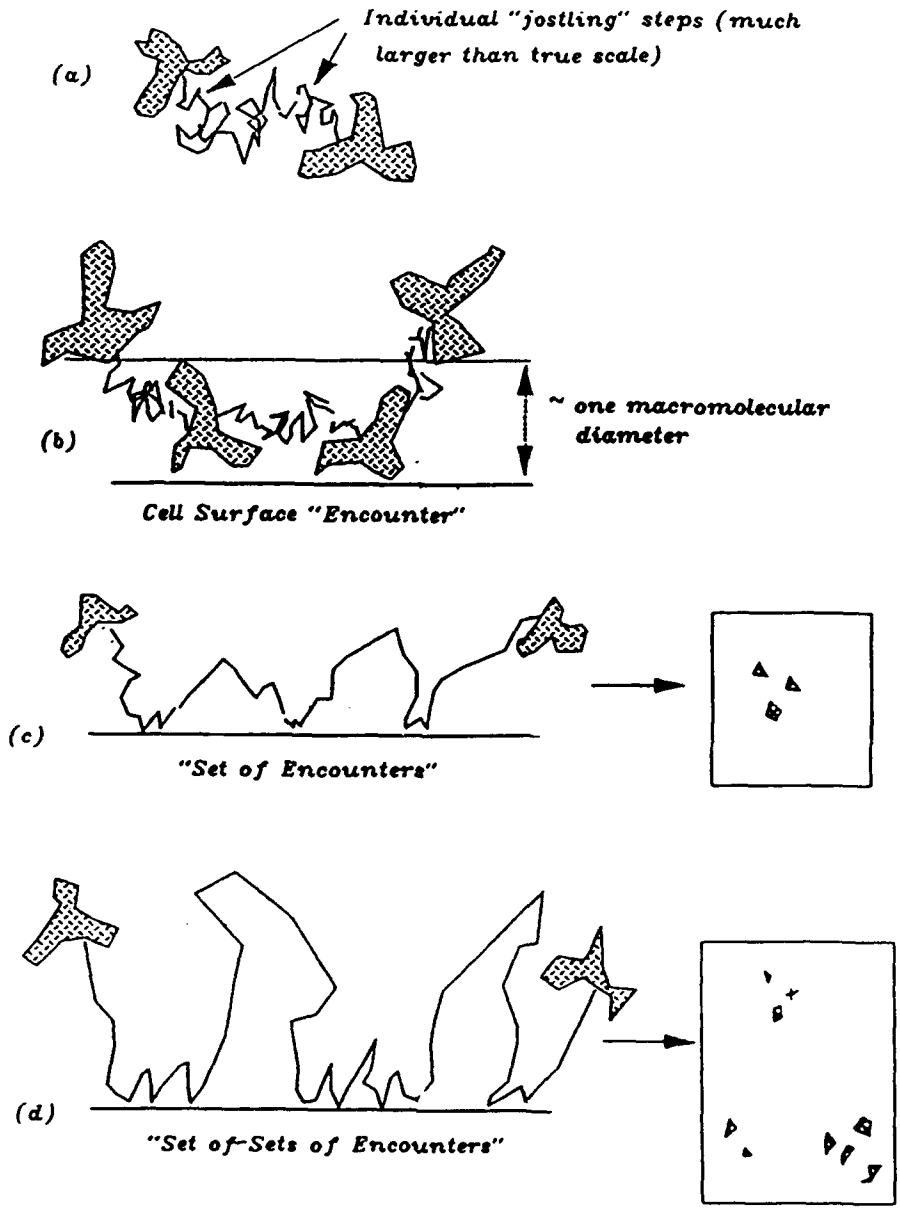


Figure 12. The hierarchy of scales of macromolecular diffusion/ cell surface interaction. (a) "Jostlings" are small gas-phase like steps. (b) "Encounters" are larger steps as molecule moves within molecular diameter of cell. (c) "Sets of Encounters" are nearby repeat encounters. (d) "Sets of Sets of Encounters" are encounter sets separated by large excursions away from cell surface.

created the fractal object. If, for example, a comparison of two snowflakes was made, the shapes would probably be different enough to prevent a direct comparison. However the determination of fractal dimension yields a single number that does allow direct comparison. If the dimension of the two snowflakes were the same, it would at least provide additional characterization of the snowflakes, and might even provide insight into snowflake creation and development. This ability to quantify irregular and random shapes is one of the important uses and benefits of fractal mathematics.

One of the largest fields of applied fractal geometry is in the study of heterogeneous reactions. These are reactions in which the reactants are not homogeneously mixed. This type of reaction is found in reactions on boundaries between phases, and in understirred conditions such as in solids. Classical kinetics is unable to accurately describe these reactions, for they give rise to fractal-like phenomena such as anomalous fractional reaction orders and time-dependent rate "constants". (15) Fractal math makes possible the development of new theories of heterogeneous reaction kinetics that are able to describe these fractal-like phenomena.

4. The Computer Model

The computer model that is the basis of this study was designed to provide new insight into the biological system of receptor cross-linking. The model can provide a wealth of information regarding the requirements and characteristics of the system as well as any difficulties or advantages of current theories. The model will be an initial step in the process to generate a new theory, which can then be used to improve intervention into the immune system. The program was written in FORTRAN 77 on a Silicon Graphics Personal Iris workstation, and utilized the graphics capability of the computer to provide a real-time visual output.

The visual nature of the model is of great value since it allows the investigator to get an idea of what is happening in the system, and to analyze the situation as it occurs. This type of visual presentation leads to an increased understanding of the situation due to the ease of understanding information in graphical format. This visual computing model allows the presentation of more information simultaneously, and allows flexibility beyond that of a system with only numerical output.

The computer program provides both the visual output, and a numerical output that allows for computational methods of analysis. The use of a "Monte Carlo" method of random events gives each run statistical significance.

Thus both the picture on the screen, and the numbers in the output file have a certain degree of verisimilitude every time the program is run.

This model is based on the B-lymphocyte interactions with a multivalent antigen, but programing flexibility allows the program to model other simple intercellular communication systems with minor adjustments. This means that not only can the program be used to examine the differences between antigens of different valency and their binding to B-cell receptors, but it allows simple modeling of hormonal signalling as well. This is because the basic mechanism for all cellular communication is thought to be generally the same, with cell receptors being bound to the communicating agent. The B-cell system was chosen for several reasons, the largest of which is the relative ease in examining response as compared to the T-lymphocyte. This fact has also resulted in more quantitative experimental evidence being available on the B-cell than the T-cell. The physical parameters are readily obtained for the B-cell system, and there is a greater likelihood of finding experimental evidence for comparisons with the model output.

Several assumptions were made in the creation of this model. The first was in the creation of the cell surface. A section of the cell surface was flattened, and formed the matrix for the motion of cell receptors. (Figure 13) While the cell surface is varied and curved in reality, a small

A Patch on the Surface of a B-cell is Represented as a Planar Matrix

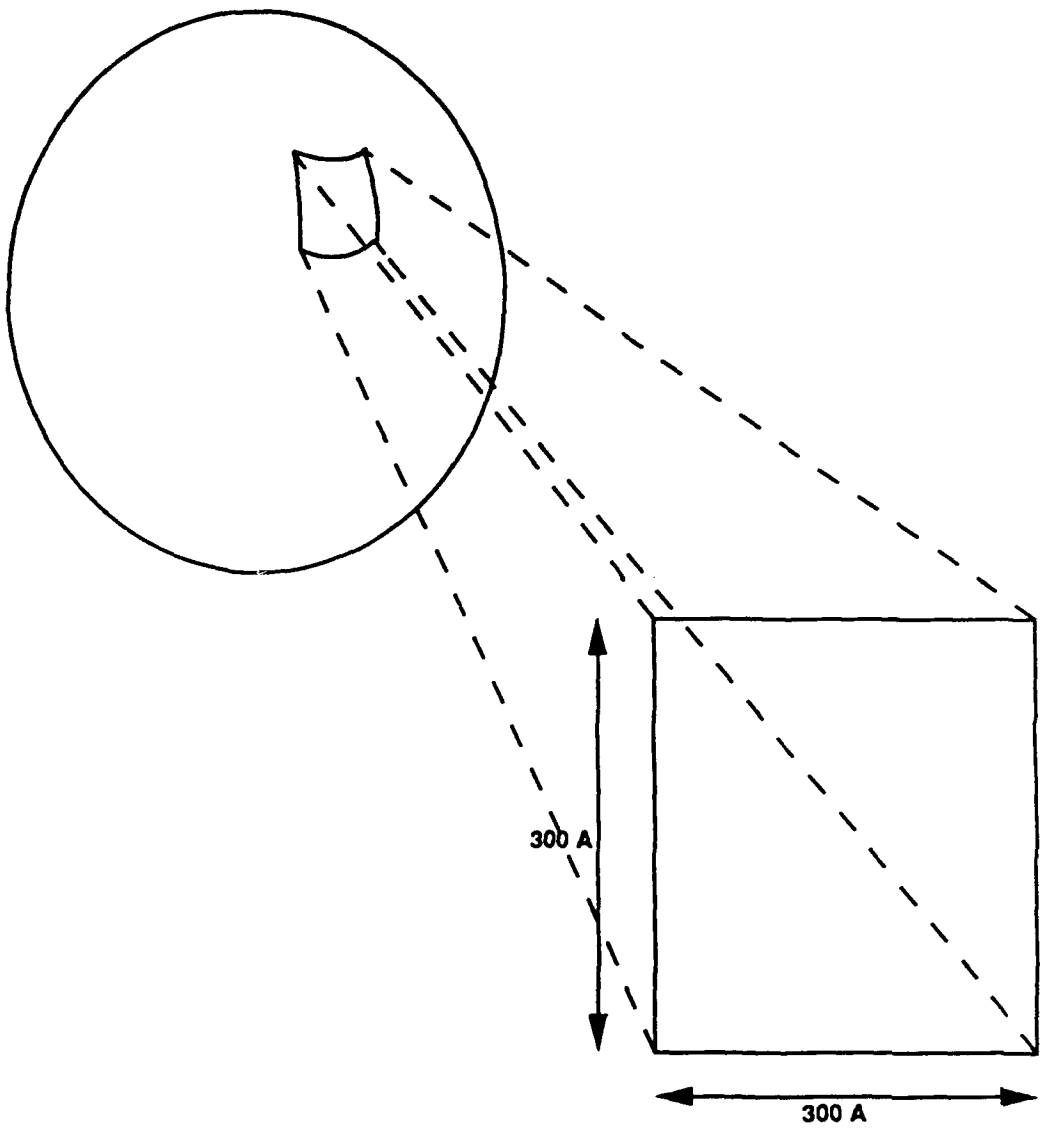


Figure 13

planar section provides a good low level approximation. The size of the patch was determined by the number of receptors. Since fifty receptors were desired for a compromise between computational tractability and model complexity, a patch with sides of 600 Angstroms was used. The following calculations show the actual values of various parameters for the patch, with initial values within acceptable limits of known values.

B-lymphocyte Radius =	8 x 10-6 m
Surface Area of B-cell =	8 x 10-10 m2
Number of B-cell Receptors =	1 x 105 per cell
Average Area Available per Receptor =	8 x 10-15 m2
Average Area for Fifty Receptors =	4 x 10-13 m2
Side of Fifty Receptor Patch =	6 x 10-7 m
Side =	6000 A

The patch was scaled down by a factor of ten in order to make the model faster and easier to observe. Decreasing the size of the cell surface patch, while keeping all other physical parameters constant, is equivalent to increasing the receptor density on the surface of the cell. An increased surface density increases the likelihood that a encounter between the antigen and the cell surface will result in an antigen-receptor bond. Since epitope separation on the antigen limits how close bond receptors can be to each other, and the size of the antigen

influences the shape of the receptor cluster, the density of receptors only acts upon the time scale of the system. Since the antigen is more likely to encounter a receptor and form a bond, the system moves faster. This allows more complete investigation of the system since letting the model run until all epitopes are bound does not take a prohibitively long period of time. Since the primary focus of the model is on structures rather than time scales, this approximation is valid and useful.

The three dimensional modeling space was bounded at the bottom with the 600 A X 600 A patch, and was extended vertically 300 angstroms to allow a limited amount of antigen motion prior to binding with receptors. The modeling space was fitted with periodic boundary conditions to keep the concentration of cell receptors and antigens constant for the entire period of the model. This means that the receptors would "wrap-around", and appear on the opposite side of the surface patch if they should wander off the patch. The same held true for the antigen molecule as well. The immunon theory was chosen for testing by the model, so the antigen was multivalent, and the receptors were bivalent with independent and separate binding sites. The immunon concept was chosen because the more popular low valence cluster theory can't explain the results of the experiments that led to the development of the immunon theory. The model hopefully will provide some insight into the reasons behind the existence of a threshold number of

antigenic epitopes. The receptors were not only bivalent, but the two binding sites were capable of some degree of independent motion, and each one was potentially active until it became bound to the antigen. The binding between antigen and receptor was irreversible to increase the probability of antigen-receptor cluster formation, and to allow for greater persistence of the cluster for observation. Since the binding constant for the cluster is very high, this approximation is valid until almost all of the epitopes on a molecule become bound and unbinding becomes possible.

The programing logic for the computer model is shown in Figure 14. The program starts by setting up the cell surface, populating it with receptors, and creating an antigen. The program then moves a receptor, followed by the antigen. When the motion is complete, the program checks to ensure the bonds of the antigen-receptor cluster have not been broken, moving the cell receptors as necessary to maintain all bonds. This simulates the antigen dragging the receptors through the plasma cell membrane as it continues on its random walk. The program then checks to see if any new bonds have been formed by receptors and epitopes moving into close proximity of each other. New bonds are formed as required, and then the program repeats the cycle by moving a receptor and the antigen. This is repeated until the desired endpoint is reached, and the investigator stops the simulation so that

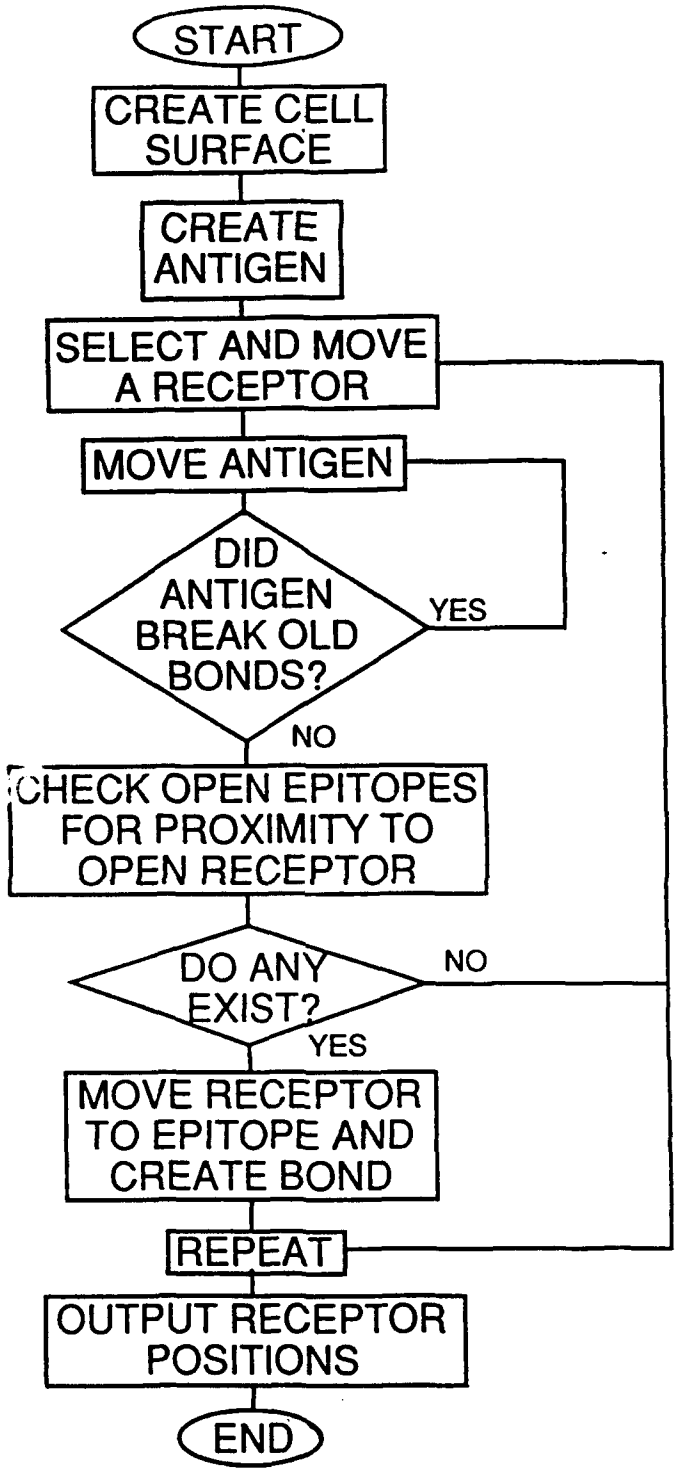


Figure 14. Program flow chart.

detailed numerical calculations and characterization of resulting cross-linked structures can be performed.

5. Detailed Run Analysis

(A) Establishing Initial Parameters

The initial portion of the program sets dimensions on all arrays. There are three main arrays used in the program. Two 19 entry arrays are used for molecular position and angle, and hold information about the antigen. The molecular position array is a 19 by 5 array. The 19 rows correspond to the 19 epitopes on the antigen, while the five columns are for the fives pieces of information about each epitope. The first three columns hold the x, y, and z coordinates of the epitope, respectively. The fourth column holds the row number of the cell receptor to which the epitope is bound, with an entry of zero indicating the absence of a bond. The final column in the molecular position array indicates the number of the binding site on the bound receptor. The molecular angle array is a 19 X 3 array, and holds the distance, angle theta, and angle phi between consecutive epitopes. The angles and distances go from the middle to the ends, with the entry for the middle epitope being unused. The angles theta and phi correspond to the angles used in spherical coordinates, with theta being the angle counterclockwise from the positive x axis, and phi being the angle down from the positive z direction.

The third array is the 50 entry cell surface array, and is used for the cell receptors. The array is three dimensional, being 50 X 3 X 3. Each of the fifty rows corresponds to a single cell receptor. The first column is the position of the Fc portion of the receptor molecule, while the second and third columns hold the locations of the two Fab binding sites of the receptor. The three entries for each receptor segment hold the x and y coordinates and a binding tag, respectively. The binding tag is zero if the receptor is unbound, and holds a number if the receptor is part of a cluster. The binding tags for the Fab binding sites are set to one if that binding site is part of an antigen-receptor bond, while the Fc binding tag is set to one if either of the Fab binding sites is bound to the antigen. This provides identification of which receptors are involved in cluster formation.

Another task of the initial section of computer code is to set up the seed values for the random number generators. These generators are not completely random, and require a four-digit seed value to start them up. The seed values are placed in a common block so that they are actually used only once. The random number generator starts with the seed and then builds a series of nearly random numbers. Each seed always generates the same series, but by using different seeds, and by placing the seed values in a common block so that the series continues rather than starting over, the numbers generated are

virtually random.

(B) Environment Initiation

After setting the dimensions on all arrays and initializing other program parameters, the program sets up the model environment. The cell surface is 600 angstroms on a side, with the origin being assigned to the middle of the surface patch. Fifty receptors are distributed randomly across the patch by using a random number generator to pick x and y coordinates of each receptor. The coordinates were allowed to vary from -250 to 250 angstroms with respect to the origin. The random number generator was used again with different seed values to get values for the separation and angular direction of the receptor's binding Fab segments. The distance between the binding sites could range from 20 to 80 angstroms, and they could be oriented at any angle to the coordinate axes so as to simulate flexibility of the immunoglobulin molecule. The locations of the binding sites were then converted from the angle and distance into x-y coordinates, and three positions were saved into the receptor array. The position where the receptor intersected the cell surface was assigned position one, while the x and y coordinates of the two binding site were assigned positions two and three in the cell surface array.

The antigen molecule was also created in this section of the program. The antigen was assigned a valency of 19,

with the epitopes numbered from one end of the molecule. This value was chosen because molecules with that valency hve been shown to be immunogenic. (8) The random number generator was used to give x, y, and z coordinates within the model environment to the center of the antigen, epitope number 10. The program then used the random number generator with different seeds to calculate the distance and angle to the next epitope using spherical coordinates. The distance between consecutive epitopes was allowed to vary between 10 and 30 angstroms, while the angle limits were varied between runs, going from no angle restrictions, to limits of 45 degrees. This simulated the varying degrees of flexibility of the polymer backbone of the antigen. These spherical coordinates (distance, theta, and phi) were placed in the molecular angle array according to the number of their epitope. The x, y, and z coordinates of the epitope are then determined by converting the angles and distances into the rectangular coordinate system. The rectangular coordinates of each epitope are assigned positions in the molecular position array. While the positions in all the arrays are filled, the corresponding binding tags are set to zero to indicate that no binding has taken place. With the positions of all receptors and epitopes complete, the program then draws a picture of the model.

(C) Graphical Presentation

The program takes the positions of the moving sites from the various arrays, and creates a visual representation of the system. The program uses a double buffer technique to create the picture on an unseen screen buffer, then place the entire picture in view at one time by switching the two buffers. Thus the screen buffer is initially blank, and after the completion of the model representation on the back buffer, the buffers are switched. placing the picture on the screen. The old picture is replaced in turn when the program puts the new picture on the back buffer, and switches buffers again, placing the new picture in view, and moving the buffer back where it can be modified.

The program first draws out the cell boundaries by lining out the cell surface patch. The sides are 600 angstroms and stretch from -300 to 300 angstroms along the x and y axes. The patch is outlined, and located at the bottom of the visual representation.

The program then places the cell receptors in the picture. The receptor center is placed directly at the cell surface, with a z coordinate of zero. The receptor is drawn by making a vertical line 20 angstroms high, and then two diagonal lines from the end of the vertical line out to the binding sites. The vertical Fc section is drawn twice as thick as the Fab arms, in order to make recognition easier. The receptor is drawn in green, and the program

repeats the drawing fifty times to account for all fifty receptors. The receptor is drawn in blue if it is bound to the antigen. This is determined by checking the binding flag on each receptor before setting the color for the receptor.

The program also draws the antigen in this section. A red line is drawn connecting all the epitopes, which are obtained from the molecular position array. The program cycles through the epitopes, assigning each one as a vertex of the line, and then draws the line from vertex to vertex. With the completion of the antigen, the visual picture is complete and the program moves it to the front buffer and it is displayed on the screen.

(D) Numerical Output

Immediately following the creation of the visual representation, the program makes a numerical output to a computer file. The program checks all nineteen epitopes on the antigen for ones that are bound to receptors. It accomplishes this by looking for the binding flags in the molecular position array. Those epitopes that are bound are examined to find the corresponding receptors. The Fc position on the corresponding receptors is then written to an output file, along with the number of bound receptors on the antigen. The output file thus contains the positions on the cell surface where bound receptors pass into the cell membrane. The output file provides data that can be

used in numerical manipulation and provides quantification of the model results as it relates to structural characteristics of cross-linked receptor clusters.

(E) Receptor Motion

The receptor motion portion of the program accounts for the plasma membrane of the cell. The receptors are moved across the cell surface to simulate the fluidity of the cell membrane. In this part of the program, a random receptor is picked with a random number generator, and then moved. The receptor is moved a distance and angle from its old position, and the x-y coordinates of its new position are calculated from the angle and distance. There is no limit on the direction of motion, but the distance the receptor moves is restricted to be between four and twelve angstroms, in order to give the model a short time scale. The new coordinates of the base are checked, and if it is less than 50 angstroms from the edge of the viewing region, the receptor is moved to the opposite side of the cell surface patch. This prevents the binding sites from extending out of the model's visual environment. The base of the receptor is moved first, then the two binding sites are moved an equal distance in the same direction.

The separation of the binding sites is also allowed to change in order to simulate conformational motion of the immunoglobulins. A random number generator is used to get a new distance and angle to the x axis for the Fab

segments, simulating rotational diffusion of the receptor. The separation distance is varied by up to 10 angstroms, but is limited to the range from 20 to 80 angstroms. The angle of the binding site to the x-axis is changed by 45 degrees. The x and y coordinates of the binding sites are then calculated using the new separation values. Following the motion of the binding sites, the coordinates for the whole receptor are written into the cell surface array.

Only one receptor is moved per iteration in order to approximately simulate motion within the plasma membrane. The membrane makes diffusion slow for the receptors than for the antigen in the surrounding medium. By moving a single receptor for each movement of the antigen, the difference between diffusional coefficients is approximated.

(F) Antigen Motion

After the cell receptor has moved, the program finds the new position of the multivalent antigen. The center of the molecule is moved up to five angstroms in a random direction. This is accomplished by using a random number generator to get changes in spherical coordinates, then converting them into the rectangular system. The center of the antigen is then compared with the visual environment boundaries, and if it is within 50 angstroms of the edge it is moved to the opposite side of the environment. This serves to keep the entire molecule together and on one side

of the visual display.

Once the new position of the antigen's center has been calculated the locations of the rest of the epitopes are found. Working out from the center, the angles and distances between epitopes are changed, making new positions for each binding site. The separation between epitopes is kept between 10 and 30 angstroms, but is varied by up to two angstroms. The angles between the epitopes, theta and phi, are changed by up to 3.6 degrees. If needed the program keeps the angle between epitopes within a certain range. The angular limits control the flexibility of the backbone of the multivalent antigen.

While calculating the new position of each epitope, the program also checks if the epitope is bound to a receptor. If an antigen-receptor bond exists at that epitope, the z coordinate of the epitope is constrained to be less than 40. This keeps bound epitopes within 40 angstroms of the cell surface. This limit maintains reasonable bond lengths between the antigen and receptors.

The motion of the molecule is difficult to perform since the program must find a movement that satisfies all of the limiting conditions. As a result, this section of the program often causes the program to slow down.

(G) Bond Maintenance

After the antigen and receptors have moved, the program ensures that the antigen is kept bound to the cell

receptors. The program checks through the molecular position array for any bound epitopes. Every time it finds a bound epitope, it compares its position to the coordinates of the receptor to which it is bound. If a discrepancy exists between the two positions, the receptor is dragged to the epitope's position. The difference in x and y coordinates between the epitope and the Fab binding arm it is bound to is applied to that Fab arm, as well as the constant region and the other binding site. This makes the binding irreversible, and allows the antigen to drag bound receptors across the cell surface. Since the plasma membranes are highly fluid in nature, the antigen would be able to drag the receptors around, although, as in the model, it would not be able to overcome the force holding the hydrophobic tail of the receptor in the cell membrane.

(H) Bond Creation

In each iteration, the program looks for new antigen-receptor bonds. The program goes through the molecular position array, and for each epitope that is not bound to a receptor, it searches the cell surface array. A comparison is made between the unbound epitope and each unbound receptor. If the distance between one of the receptor's binding sites and the epitope is less than 10 angstroms, a bond is formed. The cell receptor is dragged to the position of the epitope, and both the receptor base and the binding site involved are marked by placing the number of

the epitope involved into the cell surface array with the position of the receptor. The epitope is also marked by placing the position of the receptor in the cell surface array into the molecular position array along with the coordinates of the epitope. This section of computer code tags the bound receptor and the bound epitope with the array location of its binding partner. This allows the program to ensure all bonds stay together, and to indicate which binding sites are involved in a bond.

(I) Program Iteration

The program now goes back to the visual representation section of code, continuing with the program drawing the model environment, moving the receptors and the antigen, holding old bonds tight, and then creating new bonds. The cycle is repeated until the investigator halts the program by closing the program window. The investigator can therefore control the length time elapsed during a given run of the model, making it as long or as short as necessary. This allows the user to investigate the system at any time, and with any degree of binding. Comparisons can be made between different runs of the model at constant times or constant cluster size. The continuous iterations of the program give it flexibility by placing control over the model's duration with the investigator.

6. Results

The creation of the computer program that models the interactions between B-lymphocyte receptors and an antigen was the first, and primary, stage of this project. The second part of the project was the analysis of the data generated by multiple runs of the program. This analysis helped not only to ensure correlation of the computer model, it provided some insight into the behavior of real systems. The data generated by the computer model had two aspects, visual and numerical. As was mentioned earlier, the graphical display of the model provides insight into various aspects of the system, and can yield information that can not be easily computed or quantified. The numerical output of the program can be manipulated to provided quantified data, and can result in documentable results.

Visual Results

The first and seemingly most obvious result of the computer model is the fact that multivalent antigens can indeed cross-link receptors. Current theory raises some questions about the ability of multivalent antigens to form clusters, so this answer was not quite as obvious as one might think. One of the questions concerning the immunon theory is whether it can create a cluster with reasonable compactness. It is thought that loops of the antigen chain

would form between bound epitopes, and limit the proximity of bound receptors.

The entropy of the incoming ligand molecule is fairly high and as it becomes bound to receptors on the cell surface, it would tend to keep some portion of itself unbound and moving in order to keep this entropy. Some theories hold that the antigen would do this by forming loops off the cell surface, creating gaps between bound receptors. These gaps are thought to prevent the formation of an activating cluster by preventing the bound and activated receptors from getting close enough to each other to start the activation pathway. The model shows that another, perhaps more common method of maintaining entropy is by leaving an end portion of the antigen chain unbound. This means that instead of the portion of antigen not bound to the surface being bound to receptors at either end, the free section is bound at only one end, with the other end being free in the medium. This "free end" configuration not only can keep the same length of molecule free from the surface, and therefore quite mobile, it has a much greater range of motion. Any portion of molecular chain that is bound at both ends is limited because it must start and end at a certain point. An equal length of molecule that is only bound at one end is only constrained at that end, and therefore more movements are possible. The "free end" length of molecule has fewer constraints, and therefore has higher entropy.

The second observation concerns the motion of the antigen as it travels through the fluid medium. The molecule possesses both rotational and translational motion as it goes on its random walk in three dimensions. The interest in this is that this rotational and translational motion helps to position epitopes so that some of them are oriented towards the cell surface at all times. The motion and changes in orientation makes the possibility of binding more dependent upon the cell surface than on the antigen. Since the orientation of the Fab binding sites on the receptor can not change to any great extent, the receptor merely has to be close enough to the antigen to become involved in a bond. On the other hand, the antigen must have the proper epitope orientation and be in close proximity to the cell surface for binding to take place. At any given encounter between the cell surface and the antigen, the antigen will be positioned so that some epitopes are exposed to the surface. The antigen will even move through the duration of the encounter, which is defined as the time the antigen spends within a molecular diameter of the cell receptor height off the surface. (16) The movement will expose even more epitopes to the cell membrane. Since the motion of the antigen will expose many epitopes during the course of the cellular contact, at least one epitope will be appropriately placed for binding at every encounter. Therefore, the presence of a cell receptor at the point of encounter would be the determining

factor in the formation of a bond. The density of receptors on the cell surface, and the probability of a receptor diffusing into the area of the encounter with the antigen would determine the presence of a receptor in the contact patch. It is of interest that the model indicates that the motion of the antigen is varied enough so that the antigen can, and usually does, form a bond during every cellular encounter.

Another finding of the visual model was in the speed of binding process. The overall rate of the interaction showed a dramatic increase as more receptors became bound to cell surface receptors. Observation of the model showed that the antigen would first move through the environment, eventually approaching the cell surface. A bond with a receptor would form, and the cluster would start moving across the cell surface. The antigen would slowly move closer to the surface, and gradually form bonds with open receptors. However, when a substantial number of the epitopes on the molecule became bound, the remaining epitopes quickly made bonds to receptors. The majority of the molecule would then be on the cell surface. This would appear to correspond with denaturing of the molecule, with molecular entropy being replaced with the energy in bonds to the surface. It is an interesting result since it occurs so suddenly. Analysis shows that at a certain point enough epitopes will be bound so as to force the remainder of the molecule close to the cell surface, where binding

can occur. Since this model assumes irreversible binding, this can only lead to a large majority of the antigen's epitopes becoming involved in binding with a receptor cluster. It is of interest in real systems because it suggests that there may be a limit in the amount of binding that a molecule can sustain. Further bond formation would act to overwhelm the entropy of the molecule, and would drag the antigen down to the cell surface. Thus not only are there two types of binding, the entropic and denatured varieties, but there might be a sharp and predictable dividing line between the two.

All of the above results were also examined through the use of a crude two-dimensional model. The model was based on the three dimensional model, but was constrained to two dimensions with no moving cell receptors. This model better illustrated the rotation of the antigen as it moved close to the cell surface, and it also displayed the threshold nature of antigenic binding.

Examination of the cell receptors, both visually and through the numerical output, uncovered another interesting fact. The cell receptors tended to form bonds with only one of their two Fab binding sites. Since both binding sites are active and can form bonds, it is unusual that there are very few double-bound receptors. In the first 25 runs of the model, only one receptor became bound to the antigen with both binding sites. While this may be an artifact of the model system, it seems logical that a

double-bound receptor would be hard to create. The antigen and receptor would have to line up in just the right manner in order to get the two receptor Fab sites aligned with epitopes on the molecule. It would be far more likely for this alignment to not occur, and for the antigen to instead bind to a site on an adjacent receptor.

The final observation derived from the visual model concerns the likelihood of cross-linking in the immunon theory and the low valence cluster theory. After the initial encounter between the antigen and the cell surface, and the binding of a single epitope to a receptor, the model indicates that cross-linking would be more likely for the immunon than for the low valent antigen. This can be determined by examining the concentration of unbound epitopes near the cell surface. The greater the density of these free binding sites, the greater the chance that a second bond will form between the antigen and the cell surface receptors. Since one epitope is bound, the antigen is forced to have the adjacent epitopes also be close to the cell surface. A multivalent antigen will therefore have several free epitopes within range of the cell receptors, while a low valency antigen will only have one or two epitopes available. (Figure 15) The remainder of the epitopes needed to make the low valence cluster are still in the medium, since only one antigen has encountered the cell surface. The concentration of unbound receptors is therefore greater for a multivalent antigen, indicating

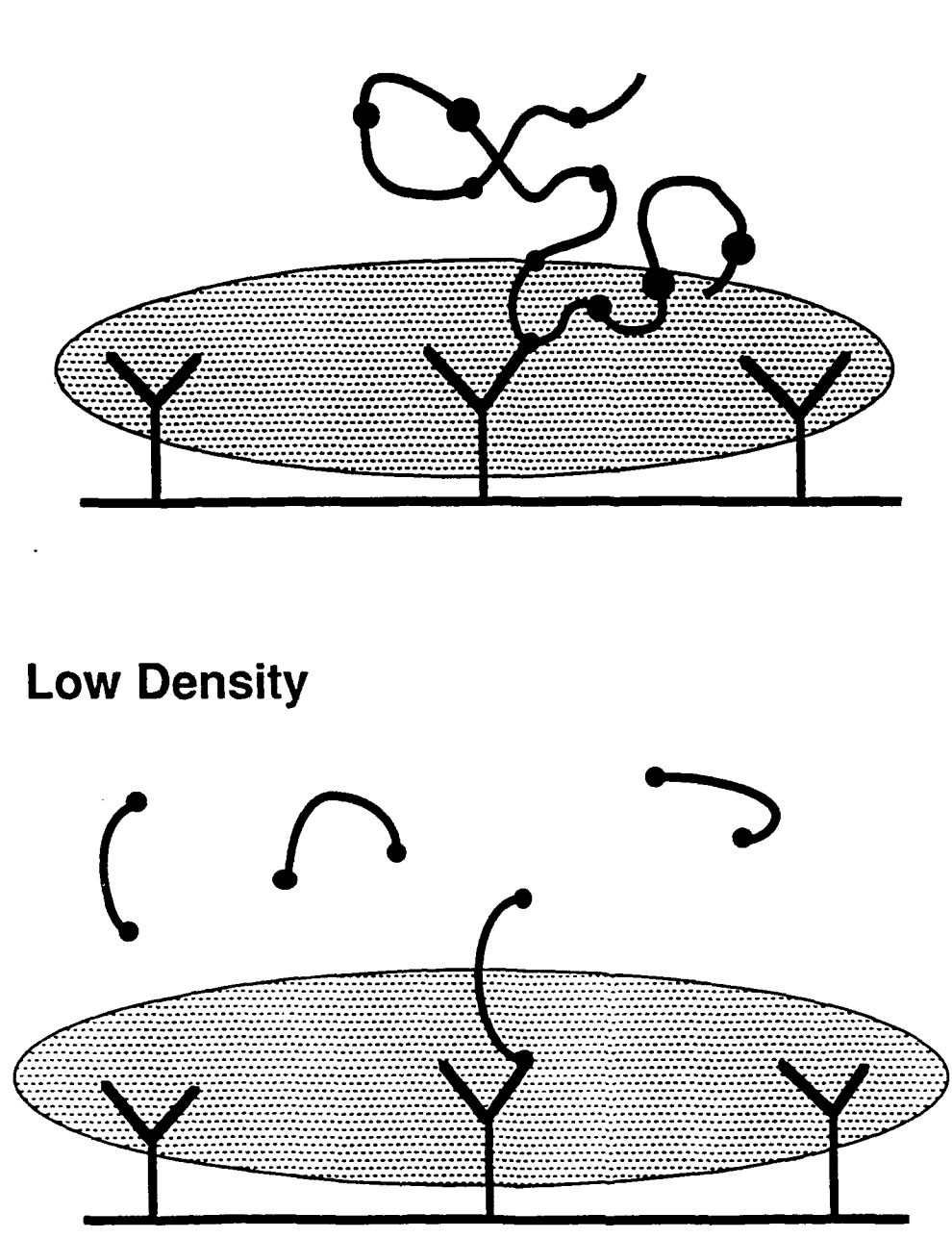


Figure 15. Relative densities of open epitopes between the two models. Top is the immunon model, while bottom shows the low valence antigen theory.

that it is more likely to cross-link receptors. This analysis holds for a single molecule encountering the cell surface, but the logic can be extended to cover a group of encounters. The extended analysis indicates that a much higher concentration of low valency antigens are required not only to match the cluster size of a multivalent molecule, but to approach its time scale of binding as well. It is inherently obvious that a greater number of low valency antigens are required to cross-link the same number of cell receptors that can be bound by a high valency antigen, but the apparently different time scale is of interest. It raises the question of which type of clustering, immunon or low valence, is more likely to stimulate the B-cell, and which is more likely to occur.

Numerical Results

The output file created by the modeling program provided quantitative information about the system, as opposed to the qualitative results gathered from the visual output. The output file contains the locations of the bases of all the receptors that are involved in bonding to the antigen. Since there is only one antigen, all these receptors are in the same antigen-receptor cluster. The base location is important because that is where the activated receptor interacts with the inside of the cell, and initiates cellular activation through interaction with intramembrane enzymes. (Figure 7) The numerical results

were analyzed for size and shape of antigen-receptor clusters. The numbers were also used to support certain trends observed visually, such as the tendency of receptors to bond with only one binding site.

The first computation to be made on the output file was the calculation of the radius of gyration for a given cluster. The radius of gyration is the radius of the circle that best fits the data points. This circle does not necessarily cover all the points, but it represents a good approximation of them all. Examination of two sample clusters, Figures 16 and 17, shows how the clusters are irregular, and better described by an ellipse than a circle. The radius is a circular approximation of this ellipse. A description of the mean square extension of the two dimensional receptor cluster is used to quantify the shape of the cluster. The square root of this extension is the radius of gyration of the object. The radius of gyration is obtained by finding the geometric mean of the eigenvalues of a 2 by 2 matrix, called the radius of gyration matrix. The matrix is defined by

$$t_{i,j} = \frac{1}{N} \sum_{k=1}^{N} (x_{ki} - \langle x \rangle_i)(x_{kj} - \langle x \rangle_j)$$

where the cluster consists of N receptors, x_1 and x_2 correspond to the x and y directions, respectively, and $\langle x \rangle$ is the mean or center of mass position,

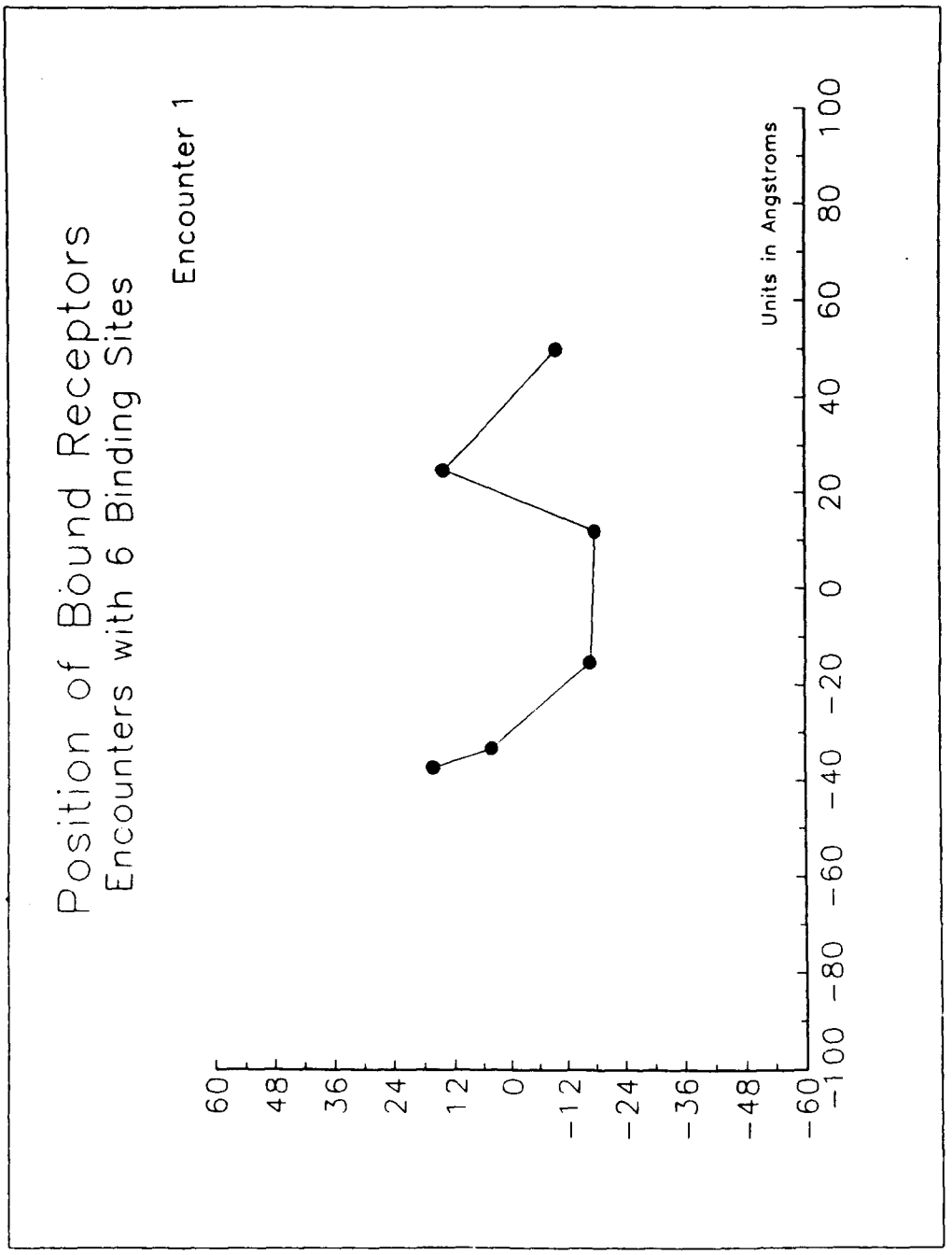


Figure 16. Diagram of antigen-receptor cluster.

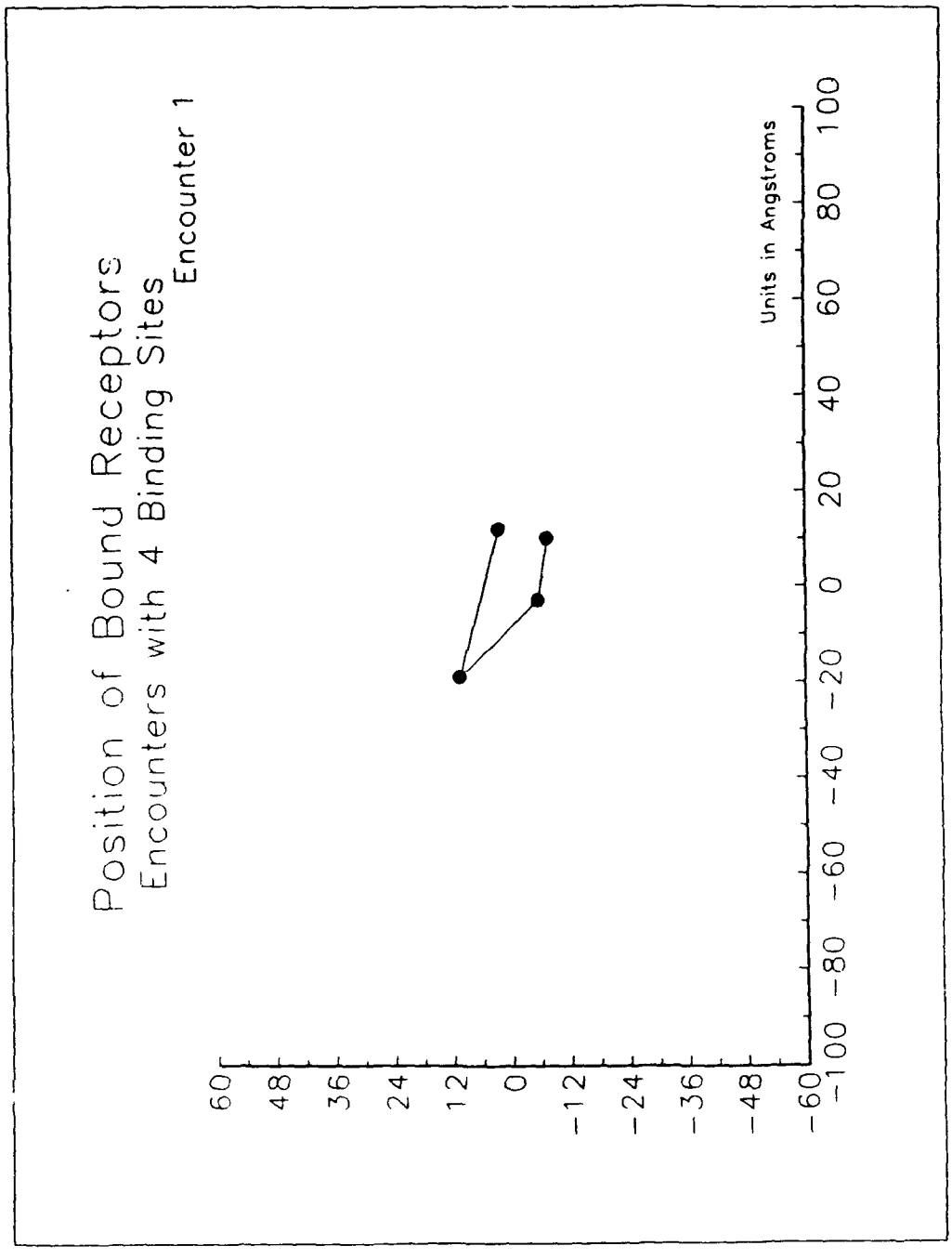


Figure 17. Diagram of antigen-receptor cluster.

$$\langle x \rangle = \frac{x_1+x_2+\ldots+x_n}{n}$$

The matrix is then filled and the eigenvalues of the matrix found by evaluation of the matrix and the solution of the resulting quadratic equation.

$$\begin{vmatrix} t_{1,1} - \alpha & t_{1,2} \\ t_{2,1} & t_{2,2} - \alpha \end{vmatrix} = 0$$

$$(t_{1,1} - \alpha)(t_{2,2} - \alpha) - t_{1,2}t_{2,1} = 0$$

$$\alpha^2 - (t_{1,1} + t_{2,2})\alpha - t_{1,2}t_{2,1} = 0$$

This yields two roots, which are the radii of gyration along the major and minor axes of the gyration ellipse. The average radius of gyration, which eliminates orientational considerations, is found by taking the geometric mean of the two directional radii:

$$r = \sqrt{r_1^2 + r_2^2}$$

The natural logarithm of the radius of gyration is then plotted against the natural logarithm of the number of points in the cluster. The slope of the line formed by several clusters of various sizes is the natural dimension of the average cluster formed. This natural dimension helps provide some quantification of random, irregular objects, and is the fractal dimension associated with the clusters formed in the model.

Twenty-five runs of the model system were analyzed at two points in the run. The first set of data was taken from early in the binding process, when several runs had not even started to bind. The clusters formed in all

25 encounters are superimposed on each other in Figure 18 to show an example of the range of possible configurations. Analysis of the clusters that had formed after 108 iterations of the program gave a fractal dimension of 2.283 ± .333. (Figure 19) This shows that the early receptor clusters are spread out fairly evenly, have equal coverage in all directions out to the limits of the cluster. Data taken later, at around 350 iterations, showed that many large clusters had formed. The fractal dimension obtained from these clusters was 1.189 ± .274. (Figure 20) These clusters become more diffuse, or elongated, as they approach the limits of their coverage. Another reason for the low dimension of these clusters is that when most of the epitopes on the linear molecule become bound, the antigen forces the receptors into a fairly linear cluster. This oblong cluster has a dimension near one, since that would be the value for a perfectly linear group of receptors. As expected, the receptor clusters average a fractal dimension between one and two. Comparison of this fractal dimension with later runs of the model, or with other physical systems may provide some insight into the interactions between cell surface receptors and antigens.

7. Conclusion

The initial purpose of this project was to create a model of the interactions between antigens and cell surface

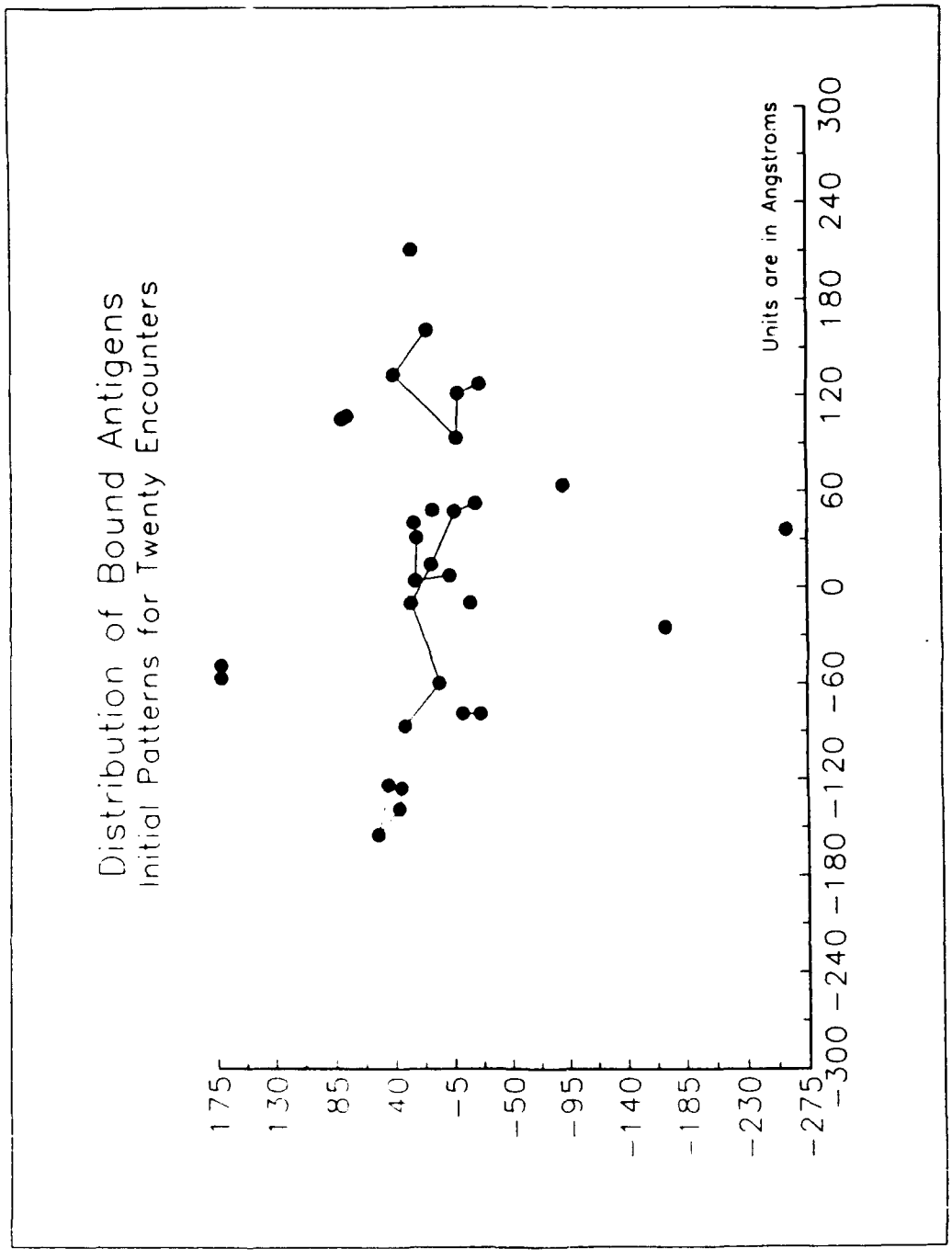

Distribution of Bound Antigens
Initial Patterns for Twenty Encounters
175
130
85
40
-5
-50
-95
-140
-185
-230
-275
-300 -240 -180 -120 -60 0 60 120 180 240 300
Units are in Angstroms

Figure 18

Determination of Fractal Dimension

Using Radius of Gyration

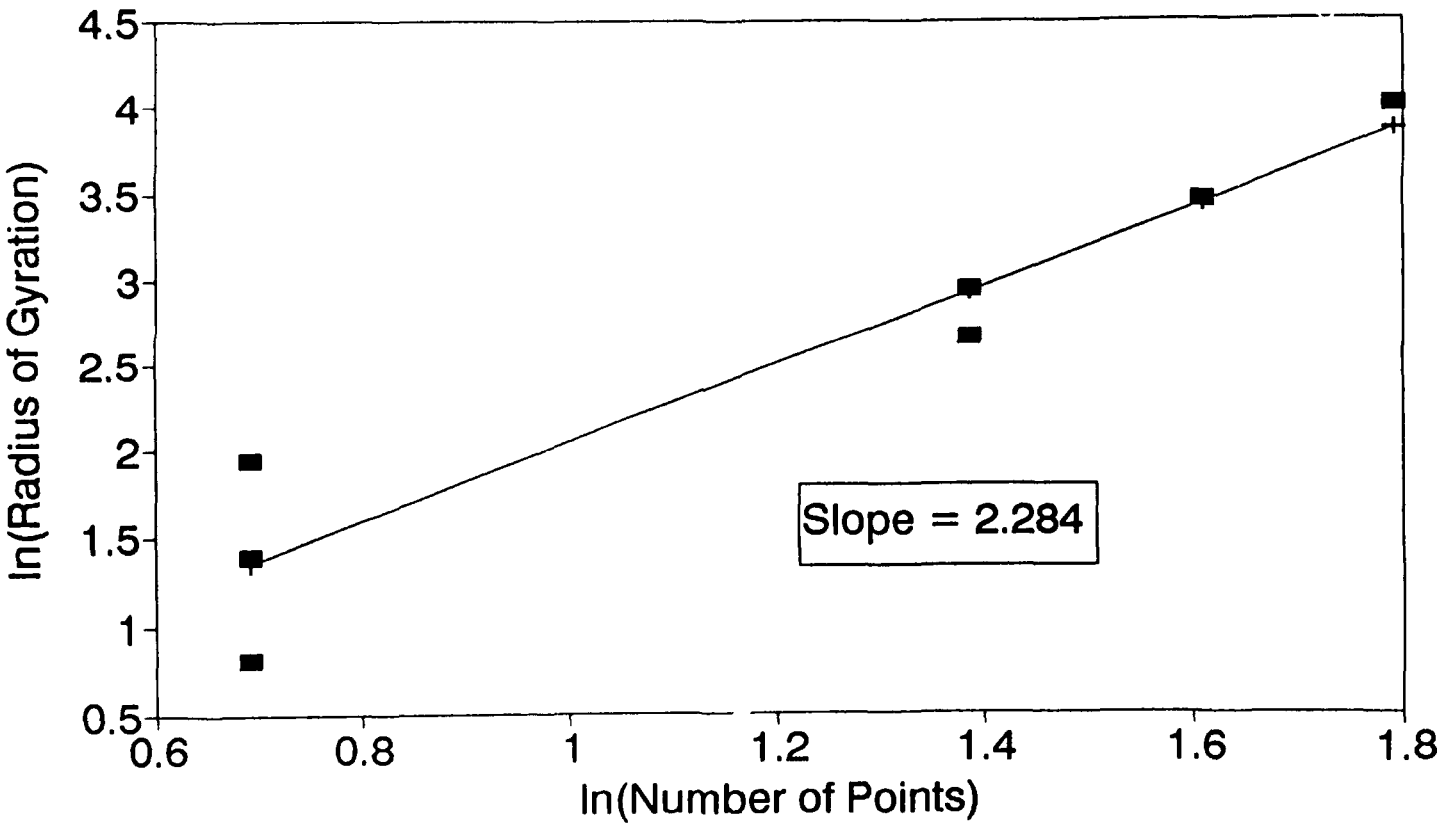


Figure 19

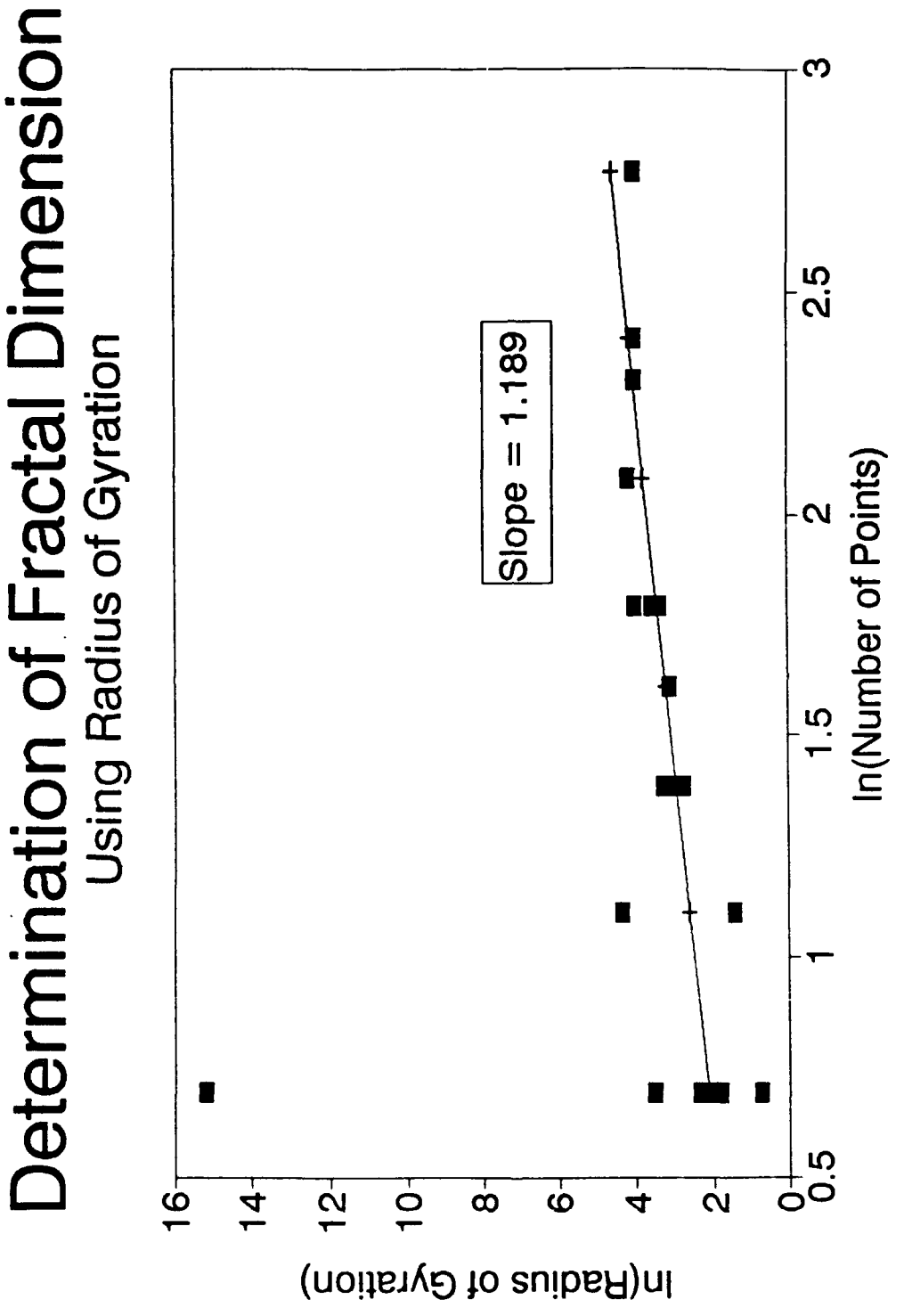

Determination of Fractal Dimension
Using Radius of Gyration
Slope = 1.189
ln(Radius of Gyration)
ln(Number of Points)
0
2
4
6
8
10
12
14
16
0.5
1
1.5
2
2.5
3

Figure 20

immunoglobulin molecules that can realistically recreate the physical system. This has been accomplished with a computer program that tested the model known as the multivalent antigen, or immunon theory. Analysis of this program has resulted in several new insights into the system, as well as numerical characterization of the antigen-receptor clusters.

The first insight is with the ability of a multivalent antigen to form compact clusters. The higher entropy of a free end as compared to a free loop appears to favor the formation of tighter clusters with antigen tails extending off the surface over the creation of more diffuse clusters with receptors separated by antigenic loops. This questions current theory which holds that loops of antigenic material between bound receptors would be very common in order to maximize the antigen's entropy.

Another interesting result was the observation of the rotational and translation motion of the antigen as it approached the cell surface. The motion of the antigen appeared to expose at least one epitope in the proper configuration for binding with every cell surface encounter. This made the probability of bond formation more dependent on the density of cell receptors than on any antigenic parameter. This is of interest because it indicates that the first contact between an antigen and a patch of cell surface with appropriate receptor density will have a high probability of resulting in the formation

of an antigen-receptor bond. This is a significant insight in understanding the apparent extremely high efficiency of intercellular communication via molecular messengers.

The third observation was the presence of a threshold in the binding of epitopes. The binding seemed to proceed at a certain rate until approximately half of the epitopes were bound to the surface. At that point a large number of the remaining free epitopes would quickly become bound to the surface, and then the binding rate would slow down again. While this corresponds in some degree to the denaturing of the antigen, it shows that the antigen has difficulty remaining free from the cell surface once a certain number of epitopes become bound.

While any of these observations might be artifacts of the model, they do raise reasonable questions about the current theory regarding the physical system. These questions encourage further investigation, both with the computer model and in actual experimentation. This project has helped in the first step along the path to the formulation of an accurate theory for describing macromolecule-cell surface receptor interactions. The road ahead is open, and the secrets of the immune system wait at the end.

REFERENCES

1. Paul, W.E., Fundamental Immunology, Raven Press, New York 1984.

2. Erhlich, P., "On Immunity with Special Reference to Cell Life", Proc. R. Soc. Lond. [Biol.], 66:424-448, 1900.

3. Herzenberg, L.A., Black, S.J., Tokuhisa, T., and Herzenberg, L.A., "Memory B-Cells at Successive Stages of Differentiation: Affinity Maturation and the Role of IgD Receptors", J. Exp, Med., 151:1071-1087, 1980.

4 Waite, B.A., and Chang, E.L., "Antibody Multivalency Effects in the Direct Binding Model for Vessicle Immunolysis Assays", J. Immunol. Methods, 115:231, 1988.

5. Parker, D.C., "Stimulation of Mouse Lymphocytes by Insoluble Anti-Mouse Immunoglobulin", Nature, 258:361-363, 1975.

6. Dintzis, R.Z., Middleton, M.H., and Dintzis, H.M., "Inhibition of Anti-Dnp Antibody Formation by High Doses of Dnp-Polyacrylamide Molecules; Effects of Hapten Density and Hapten Valence", J. Immunol., 135:423, 1985.

7. Naor, D., and Sulitzeanu, D., "Binding of Radioiodinated Bovine Serum Albumin to Mouse Spleen Cells", Nature, 214:687-688, 1967.

8. Dintzis, H.M., and Dintzis, R.Z., "A Molecular Basis for Immune Regulation: The Immunon Hypothesis", Theroretical Immunology Part One, Addison-Wesley Publishing Company, Inc., Redwood City, CA, 1987.

9. Perelson, A.S., "Paradoxes in B-Cell Stimulation by Polymeric Antigen and the Immunon Concept", Paradoxes in Immunology, CRC Press, Inc., Boca Raton, FL, 1986.

10. Parker, D.C., "Stimulation of Mouse Lymphocytes by Insoluble Anti-Mouse Immunoglobulin", Nature, 258:361-363, 1975.

11. Goldstein, B., and Dembo, M., "The IgE Mediated Activation and Desensitization of Human Basophils", Cell Surface Dymanics: Concepts and Models, Marcel Dekker, Inc., New York, 1984.

12. Perelson, A.S., "Paradoxes in B-Cell Stimulation by Polymeric Antigen and the Immunon Concept", Paradoxes in Immunology, CRC Press, Inc., Boca Raton, FL, 1986.

13. Mandelbrot, B.B., The Fractal Geometry of Nature, Freeman, San Fransisco, 1982.

14. Avnir, D., The Fractal Appraoch to Heterogeneous Chemistry: Surfaces, Colloids, and Polymers, John Wiley & Sons, New York, 1989.

15. Kopelman, R., "Fractal Reaction Kinetics", Science, 241:1620, 1988.

16. Waite, B.A., and LePage, J.F., "Toward a Monte Carlo Simulation of Receptor/Cluster Formation: The Initial Stages of Molecule/ Cell Surface Interaction", Unpublished Manuscript, 1990.

APPENDIX: COMPUTER PROGRAM

```
C     Trident Program
C     Update 5 Apr 91

C     The following include files provide graphics
C     capability.

$include /usr/include/fgl.h
$include /usr/include/fdevice.h


      real surf(50,3,3),macro(19,5),angle(19,3)
      integer mid,nbond,runner,iseed1,iseed2,iseed3,iseed4
      integer iseed5,iseed6,iseed7,iseed8,iseed9,iseed10
      integer iseed 11,iseed12
      common/array/surf(50,3,3),macro(19,5),angle(19,3),mid
      common/seeds1/iseed1,iseed2,iseed3,iseed4,iseed5,
                    iseed6
      common/seeds2/iseed7,iseed8,iseed9,iseed10,iseed11,
                    iseed12
      common/data/nbond,runner

C     This section of the program opens the graphics
C     window, sets the viewing perspective, and establishes
C     the double buffers for smoother animation.  The
C     keepas procedure makes the window square, while the
C     perspe and lookat calls create the 3-D effect by
C     assigning a point to look from and a point to look
C     at.  The final procedures set up the graphics
C     configuration of the program.

      call keepas(1,1)
      i=winope("Trident",8)
      call perspe(920,1.0,0.1,1000.0)
      call lookat(0.0,310.0,305.0,0.0,0.0,0.0,1800)
      call double
      call RGBmod
      call gconfi

C     The iseed assignments set the initial values for the
C     seeds for the random number generators used later in
C     the program.

      runner=0
      iseed1=1001
      iseed2=1010
      iseed3=1031
      iseed4=2001
      iseed5=2010
      iseed6=2031
```

```
      iseed7=3001
      iseed8=3010
      iseed9=3031
      iseed10=1321
      iseed11=1213
      iseed12=3321

C     To increase flexibility, the program makes extensive
C     use of subroutines.

      call startset
      call drawset
10    call moveset
      call tightbond
      call bondcheck
      call drawset

      runner=runner+1
      goto 10

C     The loop above is unending, so the investigator must
C     exit from the window to stop the program.  This
C     allows greater flexibility than a set end point.

      end


C     The first subroutine sets up the model environment by
C     placing the cell receptors and the antigen in their
C     initial positions.  It also starts model environment,
C     and establishes the convention that one unit of model
C     length is equivalent to one angstrom in the real
C     system.

      subroutine startset

      real pi,dis,dir,xa,ya,za,xb,yb,zb,xc,ye,zc,xt,yt,zt
      real theta,phi,one,two,three,four,ang,sep,zone,ztwo
      real yone,ytwo,xone,xtwo
      integer count,iseed1,iseed2,iseed3,iseed4,back1,back2
      integer subcount
      common/array/surf(50,3,3),macro(19,5),angle(19,3),mid
      common/seeds1/iseed1,iseed2,iseed3,iseed4,iseed5,
     1              iseed6
      common/seeds2/iseed7,iseed8,iseed9,iseed10,iseed11,
     1              iseed12
      common/data/nbond,runner

      dis=0.0
      mid=10
      nbond=0
      iiii=0
      pi=3.141592654
```

```
C     This first section finds random x and y coordinates
C     for each receptor base, and then creates a separation
C     distance and angle for the two binding sites on each
C     receptor.

      do 1020 count=1,50
        call srand(iseed1)
        xa=500.0*ran(iseed1)-250.0
        call srand(iseed2)
        ya=500.0*ran(iseed2)-250.0
        call srand(iseed3)
        dis=30.0*ran(iseed3)+10.0
        call srand(iseed4)
        dir=2.0*pi*ran(iseed4)
        xb=xa+dis*cos(dir)
        xc=xa-dis*cos(dir)
        yb=ya+dis*sin(dir)
        yc=ya-dis*sin(dir)
        surf(count,1,1)=xa
        surf(count,2,1)=ya
        surf(count,3,1)=0.0
        surf(count,1,2)=xb
        surf(count,2,2)=yb
        surf(count,3,2)=0.0
        surf(count,1,3)=xc
        surf(count,2,3)=yc
        surf(count,3,3)=0.0
1020  continue

C     The next portion finds a random position for the
C     center of the antigen.  It then goes on to compute
C     positions for the epitopes on either side of the
C     center.

      up=mid+1
      down=mid-1
      call srand(iseed7)
      macro(mid,1)=500.0*ran(iseed7)-250.0
      call srand(iseed8)
      macro(mid,2)=500.0*ran(iseed8)-250.0
      call srand(iseed9)
      macro(mid,3)=280.0*ran(iseed9)+40.0
      angle(mid,1)=20.0
      angle(mid,2)=0.0
      angle(mid,3)=0.0
      call srand(iseed10)
      theta=2.0*pi*ran(iseed10)
      call srand(iseed11)
      phi=2.0*pi*ran(iseed11)
      zone=macro(mid,3)+20.0*cos(phi)
      yone=macro(mid,2)+20.0*sin(phi)*sin(theta)
      xone=macro(mid,1)+20.0*sin(phi)*cos(theta)
```

```
      macro(down,3)=zone
      macro(down,2)=yone
      macro(down,1)=xone
      angle(down,1)=20.0
      angle(down,2)=theta
      angle(down,3)=phi
1030  call srand(iseed10)
      theta=2.0*pi*ran(iseed10)
      call srand(iseed11)
      phi=2.0*pi*ran(iseed11)
      ztwo=macro(mid,3)+20.0*cos(phi)
      ytwo=macro(mid,2)+20.0*sin(phi)*sin(theta)
      xtwo=macro(mid,1)+20.0*sin(phi)*cos(theta)
      one=sqrt(((xtwo-xone)**2)+((ytwo-yone)**2)+
               ((ztwo-zone)**2))
      if(one.lt.36.0) goto 1030
      macro(up,3)=ztwo
      macro(up,2)=ytwo
      macro(up,1)=xtwo
      angle(up,1)=20.0
      angle(up,2)=theta
      angle(up,3)=phi

C     This loop finds the position and angles for the
C     epitopes between the center and one of the ends.

      do 1060 count=12,19
        back1=count-1
        back2=count-2
        xb=macro(back1,1)
        yb=macro(back1,2)
        zb=macro(back1,3)
        xc=macro(back2,1)
        yc=macro(back2,2)
        zc=macro(back2,3)
1040    call srand(iseed12)
        dis=20.0*ran(iseed12)+10.0
        call srand(iseed10)
        theta=2.0*pi*ran(iseed10)
        call srand(iseed11)
        phi=2.0*pi*ran(iseed11)
        za=zb+dis*cos(phi)
        ya=yb+dis*sin(phi)*sin(theta)
        xa=xb+dis*sin(phi)*cos(theta)
        one=sqrt(((xb-xc)**2)+((yb-yc)**2)+((zb-zc)**2))
        two=sqrt(((xa-xb)**2)+((ya-yb)**2)+((za-zb)**2))
        three=sqrt(((xa-xc)**2)+((ya-yc)**2)+((za-zc)**2))
        four=((three**2)-(one**2)-(two**2))/(-2*one*two)
        ang=acos(four)

C     The following line limits the flexibility of the
C     molecule.
```

```
          if (ang.lt.0.5*pi) goto 1040
          macro(count,1)=xa
          macro(count,2)=ya
          macro(count,3)=za
          angle(count,1)=dis
          angle(count,2)=theta
          angle(count,3)=phi
1060    continue

C       This loop gets the positions and angles for the
C       epitopes between the center and the other end.

        do 1090 count=8,1,-1
          back1=count+1
          back2=count+2
          xb=macro(back1,1)
          yb=macro(back1,2)
          zb=macro(back1,3)
          xc=macro(back2,1)
          yc=macro(back2,2)
          zc=macro(back2,3)
1070      call srand(iseed12)
          dis=20.0*ran(iseed12)+10.0
          call srand(iseed10)
          theta=2.0*pi*ran(iseed10)
          call srand(iseed11)
          phi=2.0*pi*ran(iseed11)
          za=zb+dis*cos(phi)
          ya=yb+dis*sin(phi)*sin(theta)
          xa=xb+dis*sin(phi)*cos(theta)
          one=sqrt(((xb-xc)**2)+((yb-yc)**2)+((zb-zc)**2))
          two=sqrt(((xa-xb)**2)+((ya-yb)**2)+((za-zb)**2))
          three=sqrt(((xa-xc)**2)+((ya-yc)**2)+((za-zc)**2))
          four=((three**2)-(one**2)-(two**2))/(-2*one*two)
          ang=acos(four)

C       The following line limits the flexibility of the
C       molecule.

          if (ang.lt.0.5*pi) goto 1070
          macro(count,1)=xa
          macro(count,2)=ya
          macro(count,3)=za
          angle(count,1)=dis
          angle(count,2)=theta
          angle(count,3)=phi
1090    continue

C       This loop initializes the binding tags by setting
C       them all to zero.

        do 1100 count=1,19
          macro(count,4)=0.0
```

```
        macro(count,5)=0.0
1100  continue

      return
      end


C     This subroutine moves a receptor and the antigen.

      subroutine moveset

      real pi,xa,ya,za,dis,dir,ang,xa1,ya1,za1,ddis,ddir
      real dtheta,dphi,xbb,ybb,zbb,sep,dx,dy,dz,xt,yt,zt
      real theta,phi,one,two,three,four,dist,xb,yb,zb,xc,yc
      real angl(19,3),mem1,mem2,mem3,xaa,yaa,zaa,bond
      integer iseed1,iseed2,iseed3,iseed4,iseed5,count,pick
      integer subcount,back1,back2,front1,front2,up,down
      integer up1,down1,spot,ii,iii
      common/array/surf(50,3,3),macro(19,5),angle(19,3),mid
      common/seeds1/iseed1,iseed2,iseed3,iseed4,iseed5,
                    iseed6
      common/seeds2/iseed7,iseed8,iseed9,iseed10,iseed11,
                    iseed12

      dis=0.0
      pi=3.141592654

C     The following section selects a random receptor,
C     moves it a random distance and direction in the x-y
C     plane, and then moves the binding sites by changing
C     the separation distance and angle.

2010  call srand(iseed1)
      pick=50*ran(iseed1)+1
      call srand(iseed5)
      dis=8.0*ran(iseed5)+4.0
      call srand(iseed6)
      dir=2.0*pi*ran(iseed6)
      xa=surf(pick,1,1)
      ya=surf(pick,2,1)
      xb=surf(pick,1,2)
      yb=surf(pick,2,2)
      xc=surf(pick,1,3)
      yc=surf(pick,2,3)
      xa1=xa+dis*cos(dir)
      ya1=ya+dis*sin(dir)
      if(xa1.gt.250.0) xa1=xa1-500.0
      if(xa1.lt.-250.0) xa1=xa1+500.0
      if(ya1.gt.250.0) ya1=ya1-500.0
      if(ya1.lt.-250.0) ya1=ya1+500.0
      xb=xa1-xa+xb
      xc=xa1-xa+xc
      yb=ya1-ya+yb
```

```
      yc=yal-ya+yc
2030  dis=sqrt(((xb-xc)**2)+((yb-yc)**2))
      if(dis.gt.80.0) dis=79.0
      if(dis.lt.20.0) dis=21.0
      dir=acos((xb-xal)/(dis/2.))
      dy=yb-yal
      if (dy.lt.0.0) dir=-1*dir
      call srand(iseed3)
      ddis=20.0*ran(iseed3)-10.0
      dis=dis+ddis
      if(dis.gt.80.0) goto 2030
      if(dis.lt.20.0) goto 2030
      call srand(iseed4)
      ddir=pi/2*ran(iseed4)-(pi/4)
      dir=dir+ddir
      dis=dis/2
      xb=xal+dis*cos(dir)
      xc=xal-dis*cos(dir)
      yb=yal+dis*sin(dir)
      yc=yal-dis*sin(dir)
      surf(pick,1,1)=xal
      surf(pick,2,1)=yal
      surf(pick,1,2)-xb
      surf(pick,2,2)=yb
      surf(pick,1,3)=xc
      surf(pick,2,3)=yc

C     Once the receptor has been moved, the program moves
C     the antigen. This is accomplished by first moving the
C     middle of the molecule a random distance and
C     direction.

      mem1=macro(mid,1)
      mem2=macro(mid,2)
      mem3=macro(mid,3)
2040  xa=mem1
      ya=mem2
      za=mem3
      xaa=macro(mid,1)
      yaa=macro(mid,2)
      zaa=macro(mid,3)
      bond=macro(mid,4)
      up=mid+1
      down=mid-1
      call srand(iseed7)
      dist=3.0*ran(iseed7)+1.0
      call srand(iseed8)
      theta=2.0*pi*ran(iseed8)
      call srand(iseed9)
      phi=2.0*pi*ran(iseed9)
      zal=zaa+dis*cos(phi)
      yal-yaa+dis*sin(phi)*sin(theta)
      xal=xaa+dis*sin(phi)*cos(theta)
```

```
      if(xal.gt.250.0) xal=xal-500.0
      if(xal.lt.-250.0) xal=xal+500.0
      if(zal.gt.250.0) zal=250.0
      if(zal.lt.10.0) zal=10.0
      if(yal.gt.250.0) yal=yal-500.0
      if(yal.lt.-250.0) yal=yal+500.0
      if(bond.gt.0.0) then
        if(zal.gt.40.0) zal=40.0
      end if
      dx=xal-xaa
      dy=yal-yaa
      dz=zal-zaa
      do 2060 count=1,19
        macro(count,1)=macro(count,1)+dx
        macro(count,2)=macro(count,2)+dy
        macro(count,3)=macro(count,3)+dz
2060  continue

C     Now the program finds new angle and distances to the
C     epitopes. A random change in distance and angle is
C     generated, the new distance and angles are generated,
C     and then converted into a position for the epitope.
C     The loop does one side of the molecule, and the ii
C     counter prevents the program from getting too bogged
C     down with trying to satisfy all of the constraints.

      ii=0
2065  if(ii.gt.50) goto 2040
      ii=ii+1
      do 2080 count=up,19
        back1=count-1
        back2=count-2
        bond=macro(count,4)
2070    call srand(iseed12)
        ddis=4.0*ran(iseed12)-2.0
        call srand(iseed10)
        dtheta=0.04*pi*ran(iseed10)-0.02*pi
        call srand(iseed11)
        dphi=0.04*pi*ran(iseed11)-0.02*pi
        dis=angle(count,1)+ddis
        theta=angle(count,2)+dtheta
        phi=angle(count,3)+dphi
        xb=macro(back1,1)
        yb=macro(back1,2)
        zb=macro(back1,3)
        za=zb+dis*cos(phi)
        ya=yb+dis*sin(phi)*sin(theta)
        xa=xb+dis*sin(phi)*cos(theta)
        if(za.lt.0.0) goto 2065
        if(bond.gt.0.0) then
          if(za.gt.40.0) goto 2065
        end if
        xbb=macro(back2,1)
```

```
          ybb=macro(back2,2)
          zbb=macro(back2,3)
          one=sqrt(((xb-xbb)**2)+((yb-ybb)**2)+((zb-zbb)**2))
          two=sqrt(((xa-xb)**2)+((ya-yb)**2)+((za-zb)**2))
          if(two.gt.30.0) goto 2070
          if(two.lt.10.0) goto 2070
          three=sqrt(((xa-xbb)**2)+((ya-ybb)**2)+
                     ((za-zbb)**2))
          four=((three**2)-(one**2)-(two**2))/(-2*one*two)
          ang=acos(four)

C     The following line limits the flexibility of the
C     molecule.

          if(ang.lt.0.5*pi) goto 2065
          macro(count,1)=xa
          macro(count,2)=ya
          macro(count,3)=za
          angl(count,1)=dis
          angl(count,2)=theta
          angl(count,3)=phi
2080  continue

C     This loop is the same as the one above, except it is
C     for the other end of the molecule.

      iii=0
2095  if(iii.gt.50) goto 2040
      iii=iii+1
      do 2120 *count=down,1,-1
        back1=count+1
        back2=count+2
        bond=macro(count,4)
2100    call srand(iseed12)
        ddis=4.0*ran(iseed12)-2.0
        call srand(iseed10)
        dtheta=0.04*pi*ran(iseed10)-0.02*pi
        call srand(iseed11)
        dphi=0.04*pi*ran(iseed11)-0.02*pi
        dis=angle(count,1)+ddis
        theta-angle(count,2)+dtheta
        phi=angle(count,3)+dphi
        xb=macro(back1,1)
        yb=macro(back1,2)
        zb=macro(back1,3)
        za=zb+dis*cos(phi)
        ya=yb+dis*sin(phi)*sin(theta)
        xa=xb+dis*sin(phi)*cos(theta)
        if(za.lt.0.0) goto 2095
        if(bond.gt.0.0) then
          if(za.gt.40.0) goto 2095
        end if
        xbb=macro(back2,1)
```

```
        ybb=macro(back2,2)
        zbb=macro(back2,3)
        one=sqrt(((xb-xbb)**2)+((yb-ybb)**2)+((zb-zbb)**2))
        two=sqrt(((xa-xb)**2)+((ya-yb)**2)+((za-zb)**2))
        if(two.gt.30.0) goto 2100
        if(two.lt.10.0) goto 2100
        three=sqrt(((xa-xbb)**2)+((ya-ybb)**2)+
                  ((za-zbb)**2))
        four=((three**2)-(one**2)-(two**2))/(-2*one*two)
        ang=acos(four)

C       The following line limits the flexibility of the
C       molecule.

        if(ang.lt.0.5*pi) goto 2095
        macro(count,1)=xa
        macro(count,2)=ya
        macro(count,3)=za
        angl(count,1)=dis
        angl(count,2)=theta
        angl(count,3)=phi
2120    continue

C       The angle array is not changed until the end of the
C       subroutine in order to limit the motion of the
C       antigen, in case the routine goes through several
C       iterations.

2125    do 2130 count=1,19
          angle(count,1)=angl(count,1)
          angle(count,2)=angl(count,2)
          angle(count,3)=angl(count,3)
2130    continue

        return
        end


C       This subroutine ensures that all bonds stay together.
C       This gives the model irreversible binding.
C       Modification of this section would change binding
C       coefficient of the model, making it closer to
C       reality.

        subroutine tightbond

        real check,scheck,xone,yone,dx,dy
        integer count,counter
        common/array/surf(50,3,3),macro(19,5),angle(19,3),mid

C       This loop checks the epitopes of the antigen, and if
C       it finds one that is involved in a bond, it finds the
C       corresponding receptor and moves the receptor to the
```

```
C     epitopes coordinates.

      do 3030 count=1,19
        check=macro(count,4)
        if (check.eq.0.0) goto 3030
        xone=macro(count,1)
        yone=macro(count,2)
        scheck=macro(count,5)
        xtwo=surf(check,1,scheck)
        ytwo=surf(check,2,scheck)
        dx=xone-xtwo
        dy=yone-ytwo
        do 3020 counter=1,3
          surf(check,1,counter)=surf(check,1,counter)+dx
          surf(check,2,counter)=surf(check,2,counter)+dy
3020    continue
3030  continue

      return
      end


C     The final calculation of each iteration is the check
C     for new bond formation. The subroutine examines each
C     epitope on the antigen, and it it is not bound to a
C     receptor, it compares its coordinates to those of all
C     unbound receptors. If a separation of less than 10
C     angstroms is found, the receptor is moved to the
C     epitopes coordinates, and a bond is formed.

      subroutine bondcheck

      real xone,yone,zone,xtwo,ytwo,ztwo,space,check,pi
      real xa,ya,za,xb,yb,zb,dx,dy,dz,dis,theta,phi
      integer count,scount,subcount,middy,up,down,backl
      common/array/surf(50,3,3),macro(19,5),angle(19,3),mid
      common/data/nbond,runner

      pi=3.141592654
      middy=0

      do 4040 count=1,19
        check=macro(count,4)
        if(check.ne.0.0) goto 4040
        xone=macro(count,1)
        yone=macro(count,2)
        zone=macro(count,3)
        do 4030 scount=1,50
          do 4020 subcount=2,3
            check=surf(scount,3,subcount)
            if(check.eq.1.0) goto 4020
            xtwo=surf(scount,1,subcount)
            ytwo=surf(scount,2,subcount)
```

```
              ztwo=40.0
              dx=xone-xtwo
              dy=yone-ytwo
              space=sqrt((dx)**2+(dy)**2+(zone-ztwo)**2)
              if(space.lt.10.0) then
                macro(count,4)=scount
                macro(count,5)=subcount
                surf(scount,3,subcount)=1.0
                surf(scount,3,1)=1.0
                middy=count
                nbond=nbond+1
                do 4010 counter=1,3
                  surf(scount,1,counter)=
                                    surf(scount,1,counter)+dx
                  surf(scount,2,counter)=
                                    surf(scount,2,counter)+dy
4010              continue
              end if
4020        continue
4030      continue
4040    continue

C       If a new bond is formed, the program makes it the
C       "middle" of the molecule. This makes motion
C       computation easier, since constraints on the motion
C       of the molecule's "middle" are easier to compute. The
C       new "middle" also requires the recalculation of all
C       values in the angle array. This is done by using the
C       rectangular coordinates of each epitope to calculate
C       the new distances and angles between epitopes.

        if(middy.eq.0) goto 4080
        if(middy.eq.mid) goto 4080
        mid=middy
        up=mid+1
        down=mid-1
        do 4050 count=up,19
          back1=count-1
          xa=macro(count,1)
          ya=macro(count,2)
          za=macro(count,3)
          xb=macro(back1,1)
          yb=macro(back1,2)
          zb=macro(back1,3)
          dx=xa-xb
          dy=ya-yb
          dz=za-zb
          dis=sqrt((dx**2)+(dy**2)+(dz**2))
          theta=atan(dy/dx)
          phi=acos(dz/dis)
          angle(count,1)=dis
          angle(count,2)=theta
          angle(count,3)=phi
```

```
4050  continue
      do 4060 count=down,1,-1
        back1=count+1
        xa=macro(count,1)
        ya=macro(count,2)
        za=macro(count,3)
        xb=macro(back1,1)
        yb=macro(back1,2)
        zb=macro(back1,3)
        dx=xa-xb
        dy=ya-yb
        dz=za-zb
        dis=sqrt((dx**2)+(dy**2)+(dz**2))
        theta=atan(dy/dx)
        phi=acos(dz/dis)
        angle(count,1=dis
        angle(count,2)=theta
        angle(count,3)=phi
4060  continue

4080  return
      end


C     The final subroutine of the program creates the
C     output of the model.  This routine uses the data in
C     the program's arrays to create a picture of the
C     system, and then selects some of that data and sends
C     it to an output file.

      subroutine drawset

      real v1(2),v2(2),v3(2),v4(2),vertex(3)
      real xb,yb,xc,yc,check,check1
      integer count
      common/array/surf(50,3,3),macro(19,5),angle(19,3),mid
      common/data/nbond,runner

C     The first section draws the box that outlines the
C     patch of cell surface involved in the model. The
C     square is 300 angstroms on a side.

      v1(1)-300.0
      v1(2)=-300.0
      v2(1)=300.0
      v2(2)=300.0
      v3(1)=-300.0
      v3(2)=300.0
      v4(1)=-300.0
      v4(2)=-300.0

      call cpack($FF000000)
      call clear
```

```
      call cpack($FFFF0088)
      call linewi(1)
      call bgnclo
      call v2f(v1)
      call v2f(v2)
      call v2f(v3)
      call v2f(v4)
      call endclo

C     This section draws the cell receptors. They are
C     created by drawing a line from the base of the
C     receptor straight up for 20 angstroms, and then two
C     diagonals out to the binding sites. A change is color
C     is used to differentiate between bound and unbound
C     receptors.

      do 5060 count=1,50
        call cpack($FF40FF00)
        check1=surf(count,3,1)
        if(check1.eq.1.0) then
          call cpack($FF4000BB)
        end if
        call linewi(6)
        call bgnlin
        vertex(1)=surf(count,1,1)
        vertex(2)=surf(count,2,1)
        vertex(3)=0.0
        call v3f(vertex)
        vertex(1)=surf(count,1,1)
        vertex(2)=surf(count,2,1)
        vertex(3)=20.0
        call v3f(vertex)
        call linewi(3)
        vertex(1)=surf(count,1,2)
        vertex(2)=surf(count,2,2)
        vertex(3)=40.0
        call v3f(vertex)
        vertex(1)=surf(count,1,1)
        vertex(2)=surf(count,2,1)
        vertex(3)=20.0
        call v3f(vertex)
        vertex(1)=surf(count,1,3)
        vertex(2)=surf(count,2,3)
        vertex(3)=40.0
        call v3f(vertex)
        call endlin
5060  continue

C     The following loop draws the antigen.  Each epitope
C     is assigned as a vertex, and a line is drawn from
C     vertex to vertex.

      call cpack($FF0000FF)
```

```
      call linewi(3)
      call bgnlin
      do 5070 count=1,19
        vertex(1)=macro(count,1)
        vertex(2)=macro(count,2)
        vertex(3)=macro(count,3)
        call v3f(vertex)
5070  continue
      call endlin
      call swapbu

C     This last section send data to an output file.  The
C     program first sends the number of bonded receptors
C     and the number of iterations, as calculated by
C     counters in the bondcheck subroutine and main
C     program, respectively.  The program then sends the x
C     and y coordinates of the constant region of every
C     cell receptor that is bound to the antigen.

      write(7,5910) nbond
5910  format(' ',i5,' bound receptors')
      write(7,5920) runner
5920  format('After ',i5,' iterations')
      do 5950 count=1,19
        check=macro(count,4)
        check1=macro(count,5)
        if(check.eq.0.0) goto 5950
        write(7,5930) surf(check,1,check1),
                      surf(check,2,check1)
5930    format(' ',2f10.5,' ')
5950  continue

      return
      end
```